# Model Counting for Logical Theories

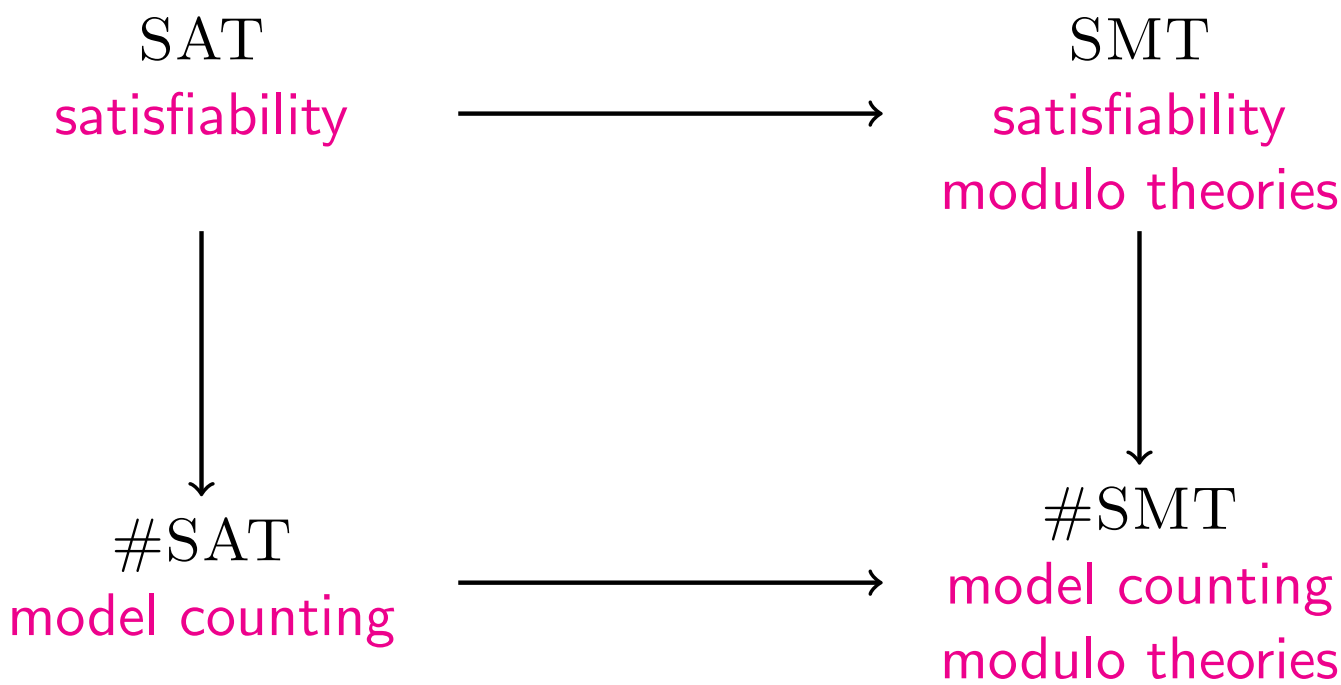
## Wednesday

Dmitry Chistikov  Rayna Dimitrova

Department of Computer Science
University of Oxford, UK

Max Planck Institute for Software Systems (MPI-SWS)
Kaiserslautern and Saarbrücken, Germany

ESSLLI 2016


SAT
satisfiability
SMT
satisfiability
modulo theories
#SAT
model counting
#SMT
model counting
modulo theories

# Agenda

| | |
|---|---|
| **Tuesday** | computational complexity, probability theory |
| **Wednesday** | randomized algorithms, Monte Carlo methods |
| **Thursday** | hashing-based approach to model counting |
| **Friday** | from discrete to continuous model counting |

# Outline

1. Randomized algorithms
   Complexity classes $\mathbf{RP}$ and $\mathbf{BPP}$

2. Monte Carlo methods

3. Markov chain Monte Carlo

# Decision problems and algorithms

Decision problem:
$L \subseteq \{0,1\}^*$ (encodings of yes-instances)

Algorithm for $L$:
says "yes" on every $x \in L$, "no" on every $x \in \{0,1\}^* \setminus L$

# Complexity classes: brief summary

$\mathbf{P}$: polynomial time (efficiently solvable)

$\mathbf{NP}$: nondeterministic polynomial time (with efficiently verifiable solutions)

$\#\mathbf{P}$: counting polynomial time

# Examples of randomized algorithms

- Primality testing
- Polynomial identity testing
- Undirected reachability
- Volume estimation

# Randomized algorithms: our model

The algorithm can toss fair coins:

1. Syntactically, a randomized algorithm is an algorithm that has access to a source of randomness, but acts deterministically if the random input is fixed.
2. It has on-demand access to arbitrary many independent random variables that have Bernoulli($\frac{1}{2}$) distribution.
3. Each request takes $1$ computational step.

# Deterministic and randomized time complexity

Recall **deterministic** time complexity:

- of algorithm $\mathcal{A}$ on input $x$

**Randomized** time complexity:
Maximum (worst-case) over all possible sequences of random bits

Then take maximum (worst-case) over all inputs of length $n$.

# Complexity classes $\mathbf{P}$, $\mathbf{RP}$, and $\mathbf{BPP}$

$\mathbf{P}$: class of languages $L$ for which there exists a deterministic polynomial-time algorithm $\mathcal{A}$ such that

$$x \in L \Longrightarrow \mathcal{A}(x) \text{ accepts}$$
$$x \notin L \Longrightarrow \mathcal{A}(x) \text{ rejects}$$

# Complexity classes $\mathbf{P}$, $\mathbf{RP}$, and $\mathbf{BPP}$

$\mathbf{P}$: class of languages $L$ for which there exists a deterministic polynomial-time algorithm $\mathcal{A}$ such that

$$x \in L \Longrightarrow \mathcal{A}(x) \text{ accepts}$$
$$x \notin L \Longrightarrow \mathcal{A}(x) \text{ rejects}$$

$\mathbf{RP}$: class of languages $L$ for which there exists a randomized polynomial-time algorithm $\mathcal{A}$ such that

$$x \in L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] \geq 1/2$$
$$x \notin L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] = 0$$

# Complexity classes $\mathbf{P}$, $\mathbf{RP}$, and $\mathbf{BPP}$

$\mathbf{P}$: class of languages $L$ for which there exists a deterministic polynomial-time algorithm $\mathcal{A}$ such that

$$x \in L \Longrightarrow \mathcal{A}(x) \text{ accepts}$$
$$x \notin L \Longrightarrow \mathcal{A}(x) \text{ rejects}$$

$\mathbf{RP}$: class of languages $L$ for which there exists a randomized polynomial-time algorithm $\mathcal{A}$ such that

$$x \in L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] \geq 1/2$$
$$x \notin L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] = 0$$

$\mathbf{BPP}$: class of languages $L$ for which there exists a randomized polynomial-time algorithm $\mathcal{A}$ such that

$$x \in L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] \geq 3/4$$
$$x \notin L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] \leq 1/4$$

# Complexity classes $\mathbf{P}$, $\mathbf{RP}$, and $\mathbf{BPP}$

Intuition:

$\mathbf{P}$: deterministic polynomial time

$\mathbf{RP}$: randomized polynomial time with one-sided error

$\mathbf{BPP}$: randomized polynomial time with bounded two-sided error

# Complexity classes $\mathbf{P}$, $\mathbf{RP}$, and $\mathbf{BPP}$

$\mathbf{P}$: class of languages $L$ for which there exists a deterministic polynomial-time algorithm $\mathcal{A}$ such that

$$x \in L \Longrightarrow \mathcal{A}(x) \text{ accepts}$$
$$x \notin L \Longrightarrow \mathcal{A}(x) \text{ rejects}$$

$\mathbf{RP}$: class of languages $L$ for which there exists a randomized polynomial-time algorithm $\mathcal{A}$ such that

$$x \in L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] \geq 1/2$$
$$x \notin L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] = 0$$

$\mathbf{BPP}$: class of languages $L$ for which there exists a randomized polynomial-time algorithm $\mathcal{A}$ such that

$$x \in L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] \geq 3/4$$
$$x \notin L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] \leq 1/4$$

# Definition of $\mathbf{RP}$ via certificates

Recall: $L \in \mathbf{NP} \iff$
there exist a polynomial $p(n)$ and a polynomial-time algorithm $V(x, y)$ such that the following holds:

$x \in L \implies$ there exists a $y \in \{0,1\}^{p(|x|)}$ such that $V(x, y) =$ YES

$x \notin L \implies$ there is no such $y \in \{0,1\}^{p(|x|)}$

$L \in \mathbf{RP} \iff$
there exists a polynomial $p(n)$ and a polynomial-time algorithm $V(x, y)$ such that the following holds:

$x \in L \implies$ for at least $\frac{1}{2}$-fraction of all $y \in \{0,1\}^{p(|x|)}$ it holds that $V(x, y) =$ YES

$x \notin L \implies$ there is no such $y \in \{0,1\}^{p(|x|)}$

# Definition of $\mathbf{RP}$ via certificates

Recall: $L \in \mathbf{NP} \iff$
there exist a polynomial $p(n)$ and a polynomial-time algorithm $V(x, y)$ such that the following holds:

$x \in L \implies$ there exists a $y \in \{0,1\}^{p(|x|)}$ such that $V(x, y) =$ YES

$x \notin L \implies$ there is no such $y \in \{0,1\}^{p(|x|)}$

$L \in \mathbf{RP} \iff$
there exists a polynomial $p(n)$ and a polynomial-time algorithm $V(x, y)$ such that the following holds:

$x \in L \implies$ for at least $\frac{1}{2}$-fraction of all $y \in \{0,1\}^{p(|x|)}$ it holds that $V(x, y) =$ YES

$x \notin L \implies$ there is no such $y \in \{0,1\}^{p(|x|)}$

Hence, $\mathbf{RP} \subseteq \mathbf{NP}$.

# Definition of $\mathbf{BPP}$ via certificates

$L \in \mathbf{RP} \iff$

there exists a polynomial $p(n)$ and a polynomial-time algorithm $V(x, y)$ such that the following holds:

$$x \in L \implies \text{for at least } \tfrac{1}{2}\text{-fraction of all } y \in \{0,1\}^{p(|x|)}$$
$$\text{it holds that } V(x, y) = \text{YES}$$

$$x \notin L \implies \text{there is no such } y \in \{0,1\}^{p(|x|)}$$

$L \in \mathbf{BPP} \iff$

there exists a polynomial $p(n)$ and a polynomial-time algorithm $V(x, y)$ such that the following holds:

$$x \in L \implies \text{for at least } \tfrac{3}{4}\text{-fraction of all } y \in \{0,1\}^{p(|x|)}$$
$$\text{it holds that } V(x, y) = \text{YES}$$

$$x \notin L \implies \text{for at most } \tfrac{1}{4}\text{-fraction of all } y \in \{0,1\}^{p(|x|)}$$
$$\text{it holds that that } V(x, y) = \text{YES}$$

# Error reduction (confidence amplification) for $\mathbf{RP}$

$\mathbf{RP}$: class of languages $L$ for which there exists a randomized polynomial-time algorithm $\mathcal{A}$ such that

$$x \in L \implies \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] \geq 1/2$$
$$x \notin L \implies \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] = 0$$

Error reduction:

- Can replace $1/2$ above with $1 - 2^{-n^d}$ for any $d \geq 1$.
- Can replace $1/2$ above with $1/n^d$ for any $d \geq 1$.

# Error reduction (confidence amplification) for $\mathbf{BPP}$

$\mathbf{BPP}$: class of languages $L$ for which there exists a randomized polynomial-time algorithm $\mathcal{A}$ such that

$$x \in L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] \geq 1 - \frac{1}{4}$$

$$x \notin L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] \leq \frac{1}{4}$$

Error reduction:

- Can replace $\frac{1}{4}$ above with $2^{-n^d}$ for any $d \geq 1$.
- Can replace $\frac{1}{4}$ above with $\frac{1}{2} - 1/n^d$ for any $d \geq 1$.

# Complexity classes $\mathbf{P}$, $\mathbf{RP}$, and $\mathbf{BPP}$

$\mathbf{P}$: class of languages $L$ for which there exists a deterministic polynomial-time algorithm $\mathcal{A}$ such that

$$x \in L \Longrightarrow \mathcal{A}(x) \text{ accepts}$$
$$x \notin L \Longrightarrow \mathcal{A}(x) \text{ rejects}$$

$\mathbf{RP}$: class of languages $L$ for which there exists a randomized polynomial-time algorithm $\mathcal{A}$ such that

$$x \in L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] \geq 1/2$$
$$x \notin L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] = 0$$

$\mathbf{BPP}$: class of languages $L$ for which there exists a randomized polynomial-time algorithm $\mathcal{A}$ such that

$$x \in L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] \geq 3/4$$
$$x \notin L \Longrightarrow \mathsf{P}[\,\mathcal{A}(x) \text{ accepts}\,] \leq 1/4$$

# Complexity classes: summary

$\mathbf{P}$: deterministic polynomial time

$\mathbf{RP}$: randomized polynomial time with one-sided error

$\mathbf{BPP}$: randomized polynomial time with bounded two-sided error

$\mathbf{NP}$: nondeterministic polynomial time (with efficiently verifiable solutions)

$\#\mathbf{P}$: counting polynomial time

# Outline

1. Randomized algorithms
   - Complexity classes $\mathbf{RP}$ and $\mathbf{BPP}$
2. Monte Carlo methods
3. Markov chain Monte Carlo

# Model counting and probability

Recall:
The model count of a formula $\varphi(x_1, \ldots, x_k)$ is $\mathsf{mc}(\varphi) = \mu([\![\varphi]\!])$.

A logical theory $\mathcal{T}$ is measured if every $[\![\varphi]\!]$ is measurable.

# Model counting and probability

Recall:
The model count of a formula $\varphi(x_1, \ldots, x_k)$ is $\mathsf{mc}(\varphi) = \mu([\![\varphi]\!])$.

A logical theory $\mathcal{T}$ is measured if every $[\![\varphi]\!]$ is measurable.

Suppose $[\![\varphi]\!] \subseteq D^k$, and assume $\mu(D) < \infty$.
Let $\mathcal{F}$ be the $\sigma$-algebra of measurable subsets of $D^k$.
Then $\mathsf{P} \colon \mathcal{F} \to [0, 1]$ given by

$$\mathsf{P}(A) = \frac{\mu(A)}{\mu(D^k)}$$

is a probability measure on $D^k$.

# Model counting and probability

Recall:
The model count of a formula $\varphi(x_1, \ldots, x_k)$ is $\mathsf{mc}(\varphi) = \mu([\![\varphi]\!])$.

A logical theory $\mathcal{T}$ is measured if every $[\![\varphi]\!]$ is measurable.

Suppose $[\![\varphi]\!] \subseteq D^k$, and assume $\mu(D) < \infty$.
Let $\mathcal{F}$ be the $\sigma$-algebra of measurable subsets of $D^k$.
Then $\mathsf{P} \colon \mathcal{F} \to [0, 1]$ given by

$$\mathsf{P}(A) = \frac{\mu(A)}{\mu(D^k)}$$

is a probability measure on $D^k$.
Then $\mathsf{mc}(\varphi) = \mathsf{P}([\![\varphi]\!]) \cdot \mu(D^k)$.

# Model counting via estimation of probability

The equality

$$\mathsf{mc}(\varphi) = \mathsf{P}([\![\varphi]\!]) \cdot \mu(D^k)$$

reduces model counting to computing the probability of an event.

# Probability estimation

Suppose we are given (implicitly) a probability space $(\Omega, \mathcal{F}, \mathrm{P})$.

Our goal is to estimate the value of $\mathrm{P}(A)$.

# Probability estimation via sampling

In order to estimate $\mathrm{P}(A)$, we can observe multiple independent copies of the indicator variable $\mathbf{1}_A$ where $A \in \mathcal{F}$.
This corresponds to random sampling and checking whether the event $A$ has occurred.

Equivalently, we observe multiple independent random variables that have Bernoulli distribution with parameter $\theta = \mathrm{P}(A)$.
Our goal is to estimate $\theta$ given the observations.

# The mean as the estimate

Given the values of $X_1, \ldots, X_n$ in $\{0, 1\}$, the unknown parameter $\theta$ should be close to

$$\overline{X} = \frac{X_1 + \ldots + X_n}{n}.$$

Why?

# The mean as the estimate

Given the values of $X_1, \ldots, X_n$ in $\{0, 1\}$, the unknown parameter $\theta$ should be close to

$$\overline{X} = \frac{X_1 + \ldots + X_n}{n}.$$

Why?

$$\begin{aligned} \mathsf{E}\,\overline{X} &= \mathsf{E}\,\frac{X_1 + \ldots + X_n}{n} = \frac{\mathsf{E}\,X_1 + \ldots + \mathsf{E}\,X_n}{n} \\ &= \frac{\theta + \ldots + \theta}{n} = \theta \end{aligned}$$

By Chebyshev's inequality, only rarely does $\overline{X}$ take values away from $\mathsf{E}\,X = \theta$.

# The mean as the estimate

Given the values of $X_1, \ldots, X_n$ in $\{0, 1\}$, the unknown parameter $\theta$ should be close to

$$\overline{X} = \frac{X_1 + \ldots + X_n}{n}.$$

Why?

$$\begin{aligned} \mathsf{E}\,\overline{X} &= \mathsf{E}\,\frac{X_1 + \ldots + X_n}{n} = \frac{\mathsf{E}\,X_1 + \ldots + \mathsf{E}\,X_n}{n} \\ &= \frac{\theta + \ldots + \theta}{n} = \theta \end{aligned}$$

By Chebyshev's inequality, only rarely does $\overline{X}$ take values away from $\mathsf{E}\,X = \theta$.

Is that sufficient?

# Approximation

We want to estimate a certain quantity $f$.

Suppose our estimate is $\tilde{f} = \tilde{f}(\varepsilon)$,
where $\varepsilon$ is an input parameter.

Additive error:

$$|\tilde{f}(\varepsilon) - f| \leq \varepsilon$$

Multiplicative error:

$$|\tilde{f}(\varepsilon) - f| \leq \varepsilon \cdot f$$

# Randomized approximation

We want to estimate a certain quantity $f$.

Suppose our estimate is $\tilde{f} = \tilde{f}(\varepsilon, \alpha)$,
where $\varepsilon$, $\alpha$ are input parameters.

Additive error:

$$\mathsf{P}\big[\, |\tilde{f}(\varepsilon, \alpha) - f| \leq \varepsilon \,\big] \geq 1 - \alpha$$

Multiplicative error:

$$\mathsf{P}\big[\, |\tilde{f}(\varepsilon) - f| \leq \varepsilon \cdot f \,\big] \geq 1 - \alpha$$

# Randomized approximation

We want to estimate a certain quantity $f$.

Suppose our estimate is $\tilde{f} = \tilde{f}(\varepsilon, \alpha)$,
where $\varepsilon$, $\alpha$ are input parameters.

Additive error:

$$\mathsf{P}\big[\, |\tilde{f}(\varepsilon,\alpha) - f| \le \varepsilon \,\big] \ge 1 - \alpha$$

Multiplicative error:

$$\mathsf{P}\big[\, |\tilde{f}(\varepsilon) - f| \le \varepsilon \cdot f \,\big] \ge 1 - \alpha$$

We want to find efficient randomized approximation schemes.

# The mean as the estimate

Given the values of $X_1, \ldots, X_n$ in $\{0, 1\}$, the unknown parameter $\theta$ should be close to

$$\overline{X} = \frac{X_1 + \ldots + X_n}{n}.$$

Why?

$$\begin{aligned} \mathsf{E}\,\overline{X} &= \mathsf{E}\,\frac{X_1 + \ldots + X_n}{n} = \frac{\mathsf{E}\,X_1 + \ldots + \mathsf{E}\,X_n}{n} \\ &= \frac{\theta + \ldots + \theta}{n} = \theta \end{aligned}$$

By Chebyshev's inequality, only rarely does $\overline{X}$ take values away from $\mathsf{E}\,X = \theta$.

# Estimates from the Chebyshev bound

Chebyshev inequality:

$$\mathsf{P}\big(\,|X - \mathsf{E}\,X| \geq t\,\big) \leq \frac{\mathsf{Var}\,X}{t^2}$$

# Estimates from the Chebyshev bound

Chebyshev inequality:

$$\mathsf{P}\big(\,|X - \mathsf{E}\,X| \geq t\,\big) \leq \frac{\mathsf{Var}\,X}{t^2}$$

$n \geq \dfrac{1}{4\alpha\varepsilon^2}$ samples are sufficient

for additive error $\varepsilon$ and confidence parameter $\alpha$

# Estimates from the Chernoff bound

Chernoff bound:

$$\mathsf{P}\big(\,|\overline{X} - \mathsf{E}\,X| \geq t\,\big) \leq 2\exp(-nt^2/4)$$

if $X_1, \ldots, X_n$ are independent and identically distributed in $[0, 1]$

# Estimates from the Chernoff bound

Chernoff bound:

$$\mathsf{P}\big(\,|\overline{X} - \mathsf{E}\,X| \geq t\,\big) \leq 2\exp(-nt^2/4)$$

if $X_1, \ldots, X_n$ are independent and identically distributed in $[0, 1]$

$n \geq \dfrac{4}{\ln(2/\alpha)\varepsilon^2}$ samples are sufficient

for additive error $\varepsilon$ and confidence parameter $\alpha$

# Conclusion: Model counting via Monte Carlo

Algorithm:

1. Given a formula $\varphi(x_1, \ldots, x_k)$, sample uniformly from possible models.
2. Return the proportion of actual models times $\mu(D^k)$.

# Outline

1. Randomized algorithms
   - Complexity classes $\mathbf{RP}$ and $\mathbf{BPP}$
2. Monte Carlo methods
3. Markov chain Monte Carlo

# The Knapsack Problem revisited

Consider the counting version of the Knapsack problem.

Given $a_1, \ldots, a_n \in \mathbb{N}$ and $b \in \mathbb{N}$, **compute the number** $N$ of vectors $(x_1, \ldots, x_n) \in \{0,1\}^n$ that satisfy $\sum_{i=1}^{n} a_i x_i \leq b$.

This problem is $\#\mathbf{P}$-complete.

# A Monte Carlo algorithm for Knapsack

$C = 0$

For $k = 1, \ldots, m$

1. Sample $(x_1, \ldots, x_n)$ uniformly at random from $\{0,1\}^n$.
2. If $\sum_{i=0}^{n} a_i x_i \leq b$ then $C = C + 1$.

Return $Y = (\frac{C}{m})2^n$

$Y$ is the random variable corresponding to the output. $\mathsf{E}(Y) = N$.

# A Monte Carlo algorithm for Knapsack

$C = 0$

For $k = 1, \ldots, m$

1. Sample $(x_1, \ldots, x_n)$ uniformly at random from $\{0, 1\}^n$.
2. If $\sum_{i=0}^{n} a_i x_i \leq b$ then $C = C + 1$.

Return $Y = (\frac{C}{m})2^n$

$Y$ is the random variable corresponding to the output. $\mathsf{E}(Y) = N$.

We can make $m$ sufficiently large to obtain a reliable approximation with any desired accuracy.

What's the problem?

# A Monte Carlo algorithm for Knapsack

$C = 0$

For $k = 1, \ldots, m$

1. Sample $(x_1, \ldots, x_n)$ uniformly at random from $\{0,1\}^n$.
2. If $\sum_{i=0}^{n} a_i x_i \leq b$ then $C = C + 1$.

Return $Y = (\frac{C}{m})2^n$

$Y$ is the random variable corresponding to the output. $\mathsf{E}(Y) = N$.

Let $Z_k$ be random variable such that $Z_k = 1$ if $\sum_{i=0}^{n} a_i x_i \leq b$ at the $k$-th iteration and $Z_k = 0$ otherwise.
$Z_1, Z_2, \ldots$ are independent Bernoulli random variables with parameter $p = \frac{N}{2^n}$. Let $\overline{Z}$ be the number of failures before the first success. $\overline{Z}$ has geometric distribution with parameter $p$. Thus,

$$\mathsf{E}\,\overline{Z} = \frac{1-p}{p} = \frac{1}{p} - 1 = \frac{2^n}{N} - 1.$$

If $N$ is sub-exponential (e.g., polynomial in $n$), we need an exponential number of steps before the first success!

# An alternative?

Sample from the uniform distribution over the set of solutions

$$\Omega_{\text{KNAPSACK}} = \{(x_1, \ldots, x_n) \in \{0,1\}^n : \sum_{i=0}^{n} a_i x_i \leq b\}.$$

How? Use a Markov chain Monte Carlo method.

# Markov Chains: Definition

**Finite Markov chain** $\mathfrak{M} = (\Omega, T)$

- finite set of states $\Omega$,
- transition probability matrix $T$ where

$$T_{s,s'} = \mathsf{P}(\text{next state will be } s' \mid \text{ the current state is } s)$$

# Markov Chains: Definition

**Finite Markov chain** $\mathfrak{M} = (\Omega, T)$

- finite set of states $\Omega$,
- transition probability matrix $T$ where

$$T_{s,s'} = \mathsf{P}(\text{next state will be } s' \mid \text{ the current state is } s)$$

A Markov chain $\mathfrak{M}_{\text{KNAPSACK}} = (\Omega_{\text{KNAPSACK}}, T_{\text{KNAPSACK}})$ for Knapsack

- $\Omega_{\text{KNAPSACK}} = \{(x_1, \ldots, x_n) \in \{0,1\}^n \colon \sum_{i=1}^n a_i x_i \leq b\}$,
- transition probability matrix $T_{\text{KNAPSACK}}$ has the following rules for transitioning from $s = (x_1, \ldots, x_n)$ to $s' = (y_1, \ldots, y_n)$
  1. Set $s' = s$ with probability $\frac{1}{2}$.
  2. Select $i$ uniformly at random from $\{1, \ldots, n\}$ and let

$$\overline{s} = (x_1, \ldots, x_{i-1}, 1 - x_i, x_{i+1}, \ldots, x_n).$$

  3. If $\sum_{i=1}^n a_i \overline{s}_i \leq b$, then set $s' = \overline{s}$ otherwise $s' = s$.

# Markov chain for the Knapsack problem

$\mathfrak{M}_{\text{KNAPSACK}} = (\Omega_{\text{KNAPSACK}}, T_{\text{KNAPSACK}})$ for Knapsack

- $\Omega_{\text{KNAPSACK}} = \{(x_1, \ldots, x_n) \in \{0,1\}^n \colon \sum_{i=1}^n a_i x_i \leq b\}$,
- transition probability matrix $T_{\text{KNAPSACK}}$ has the following rules for transitioning from $s = (x_1, \ldots, x_n)$ to $s' = (y_1, \ldots, y_n)$
  1. Set $s' = s$ with probability $\frac{1}{2}$.
  2. Select $i$ uniformly at random from $\{1, \ldots, n\}$ and let
  $$\overline{s} = (x_1, \ldots, x_{i-1}, 1 - x_i, x_{i+1}, \ldots, x_n).$$
  3. If $\sum_{i=1}^n a_i \overline{s}_i \leq b$, then set $s' = \overline{s}$ otherwise $s' = s$.

Property:

The transition probability matrix of $\mathfrak{M}_{\text{KNAPSACK}}$ is symmetric.

$$T_{u,v} = \frac{1}{2} \cdot \frac{1}{n} = T_{v,u} \text{ for } v \neq u.$$

# Simulating a Markov chain

Consider a Markov chain $\mathfrak{M} = (\Omega, T)$.

Let $D : \Omega \to \mathbb{R}$ be a probability distribution on $\Omega$.

Interpreting $D$ as an element of $\mathbb{R}^{|\Omega|}$ we have that $D' = DT$ is also a probability distribution over $\Omega$ (since $\sum_{s' \in \Omega} T_{s,s'} = 1$).

Suppose we start in some state $s_0$ and simulate the Markov chain. Let $X_t$ be the random variable such that

$$X_t = s \text{ iff the current state at step } t \text{ is } s.$$

For the corresponding sequence of distributions $D[0], D[1], D[2], \ldots$ over $\Omega$ we have that

- $D[0](s_0) = 1$ and $D_0(s) = 0$ for all $s \neq s_0$ and
- $D[i+1] = D[i]T$ for all $i \geq 0$.

# Markov Chains: Stationary distribution

**Stationary distribution** of a Markov chain $\mathfrak{M} = (\Omega, T)$ is a probability distribution $D : \Omega \to \mathbb{R}$ such that $DT = D$, that is,

$$\sum_{u \in \Omega} D_u \cdot T_{u,v} = D_v \text{ for all } v \in \Omega.$$

# Markov Chains: Stationary distribution

**Stationary distribution** of a Markov chain $\mathfrak{M} = (\Omega, T)$ is a probability distribution $D : \Omega \to \mathbb{R}$ such that $DT = D$, that is,

$$\sum_{u \in \Omega} D_u \cdot T_{u,v} = D_v \text{ for all } v \in \Omega.$$

Does every finite Markov chain have a stationary distribution?

# Markov Chains: Stationary distribution

**Stationary distribution** of a Markov chain $\mathfrak{M} = (\Omega, T)$ is a probability distribution $D : \Omega \to \mathbb{R}$ such that $DT = D$, that is,

$$\sum_{u \in \Omega} D_u \cdot T_{u,v} = D_v \text{ for all } v \in \Omega.$$

Does every finite Markov chain have a stationary distribution? Yes.

Every stochastic matrix always has an eigenvalue of $1$ with a left eigenvector whose entries are nonnegative.

Not true in general for Markov chains with infinite state space.

# Markov Chains: Stationary distribution

**Stationary distribution** of a Markov chain $\mathfrak{M} = (\Omega, T)$ is a probability distribution $D : \Omega \to \mathbb{R}$ such that $DT = D$, that is,

$$\sum_{u \in \Omega} D_u \cdot T_{u,v} = D_v \text{ for all } v \in \Omega.$$

Does every finite Markov chain have a unique stationary distribution?

# Markov Chains: Stationary distribution

**Stationary distribution** of a Markov chain $\mathfrak{M} = (\Omega, T)$ is a probability distribution $D : \Omega \to \mathbb{R}$ such that $DT = D$, that is,

$$\sum_{u \in \Omega} D_u \cdot T_{u,v} = D_v \text{ for all } v \in \Omega.$$

Does every finite Markov chain have a unique stationary distribution?
No.

Example
Take $\mathfrak{M} = (\Omega, T)$ with $\Omega = \{u, v\}$ and $T_{u,u} = 1$ and $T_{v,v} = 1$.
Every probability distribution $D$ on $\Omega$ is a stationary distribution.

# Stationary distribution for the Knapsack problem

The uniform distribution $U$ over $\Omega_{\text{KNAPSACK}}$ is a stationary distribution for $\mathfrak{M}_{\text{KNAPSACK}} = (\Omega_{\text{KNAPSACK}}, T_{\text{KNAPSACK}})$.

We have to show that $\sum_{s \in \Omega_{\text{KNAPSACK}}} U_s \cdot T_{s,v} = U_v$ for all $v$.

Consider $v \in \Omega_{\text{KNAPSACK}}$ and suppose $v$ has $d$ "neighbours" different from $v$. Recall that $T_{s,s'} = T_{s',s} = \frac{1}{2n}$ for $u \neq v$. We have

$$\begin{aligned} \textstyle\sum_s U_s \cdot T_{s,v} &= U_v \cdot T_{v,v} + \textstyle\sum_{s, s \neq v} U_s \cdot T_{s,v} \\ &= \tfrac{1}{N}(\tfrac{1}{2} + \tfrac{1}{2}(1 - \tfrac{d}{n})) + d\tfrac{1}{N}\tfrac{1}{2n} \\ &= \tfrac{1}{N} = U_v \end{aligned}$$

# Irreducible Markov chains

A **path** in $\mathfrak{M} = (\Omega, T)$ is a sequence of states $s_0, s_1 \dots s_l$ such that $T_{s_i, s_{i+1}} > 0$ for $0 \leq i < l$. We say that $s_l$ is **reachable** from $s_0$.

In the Markov chain $\mathfrak{M}_{\text{KNAPSACK}}$

- if we start from $(0, \dots, 0)$ we can reach any state,
- if we start from any state we can reach $(0, \dots, 0)$.

Thus, from any state of $\mathfrak{M}_{\text{KNAPSACK}}$ we can reach any state.
Such Markov chains are called **irreducible**.

Recall the example with no unique stationary distribution.

# What about convergence?

If a Markov chain has a unique stationary distribution $D$, does it always converge to $D$, starting from any distribution?
That is, for any $D[0]$, does the sequence $D[0], D[1], D[2], \ldots$, where $D[i+1] = D[i]T$, converge to $D$?
Not necessarily.

## Example

Take $\mathfrak{M} = (\Omega, T)$ with $\Omega = \{u, v\}$ and $T_{u,v} = 1$ and $T_{v,u} = 1$.
The only stationary distribution is $D(u) = D(v) = \frac{1}{2}$.
Now, take $D[0]$ such that $D[0](u) = 1$ and $D[0](v) = 0$.

The reason for which the Markov chain does not converge is that the state periodically alternates between states $u$ and $v$.

# Aperiodic Markov chains

A state $s$ of a Markov chain is called **periodic** if there exists $k \in \mathbb{N}$, $k > 1$ such that for every path $s = s_0, \ldots, s_l = s$, $k$ divides $l$.

A state which is not periodic is called aperiodic. A Markov chain is called **aperiodic** if all of its states are aperiodic.

# Aperiodic Markov chains

A state $s$ of a Markov chain is called **periodic** if there exists $k \in \mathbb{N}$, $k > 1$ such that for every path $s = s_0, \ldots, s_l = s$, $k$ divides $l$.

A state which is not periodic is called aperiodic. A Markov chain is called **aperiodic** if all of its states are aperiodic.

Is $\mathfrak{M}_{\text{KNAPSACK}}$ aperiodic?

# Aperiodic Markov chains

A state $s$ of a Markov chain is called **periodic** if there exists $k \in \mathbb{N}$, $k > 1$ such that for every path $s = s_0, \ldots, s_l = s$, $k$ divides $l$.

A state which is not periodic is called aperiodic. A Markov chain is called **aperiodic** if all of its states are aperiodic.

Is $\mathfrak{M}_{\text{KNAPSACK}}$ aperiodic? Yes.

For every state $s \in \Omega_{\text{KNAPSACK}}$, $T_{s,s} > 0$. Thus there exists a path from $s$ to $s$ of arbitrary length, and thus every state is aperiodic.

# Fundamental Theorem of Markov chains

**Theorem** Every finite Markov chain $\mathfrak{M}$ which is irreducible has a unique stationary distribution $D$, and if $\mathfrak{M}$ is aperiodic, then it also holds that $\lim_{i\to\infty} D[i] = D$.

This means that the Markov chain $\mathfrak{M}_{\text{KNAPSACK}}$ converges to the uniform distribution over $\Omega_{\text{KNAPSACK}}$.

# Using $\mathfrak{M}_{\text{KNAPSACK}}$

Almost uniform sampling for Knapsack using $\mathfrak{M}_{\text{KNAPSACK}}$.

1. Start in state $s = (0, \ldots, 0)$.
2. Simulate $\mathfrak{M}_{\text{KNAPSACK}}$ for sufficiently many steps until the distribution over states is "close" to the uniform distribution over $\Omega_{\text{KNAPSACK}}$.
3. Return the current state.

Repeating this we can obtain a sequence of independent samples. Uniform sampling from $\Omega_{\text{KNAPSACK}}$ can be used to obtain a randomized approximation algorithm for the counting the number of solutions to the Knapsack problem.

How many steps is sufficiently many?

It is not known if $\mathfrak{M}_{\text{KNAPSACK}}$ converges to the uniform distribution in polynomial number of steps. ($\Omega$ may be exponential in $n$.)

# Markov chain Monte Carlo

Markov chain Monte Carlo (MCMC) is a technique for sampling from a complicated distribution using local information.

The main challenge is to obtain good bounds on the number of steps a Markov chain takes to converge to the desired distribution.

MCMC may provide efficient (i.e., polynomial time) solution techniques.

# Computing the volume of a convex body

Given a convex body $K \subseteq \mathbb{R}^n$, compute its volume $Vol(K)$.

The computational effort required increases as $n$ increases.

[Dyer and Frieze'88] Computing the volume exactly is $\#\mathbf{P}$-hard.

[Dyer, Frieze and Kannan'91] Polynomial randomized approximation algorithm via Markov chain Monte Carlo.

# Input to the algorithm

$K$ is given as a membership oracle.

Two $n$-dimensional balls $B_0 \subseteq K \subseteq B_r$ of non-zero radius.

By simple transformations of $K$ it can be ensured that $B_0$ is the unit ball and that $B_r$ has radius $cn \log n$ for some constant $c$.

Note: The volume of the smallest ball containing $K$ might be exponential in $Vol(K)$, hence naive Monte Carlo is hopeless.

# From volume computation to uniform sampling

Construct a sequence of concentric balls
$B_0 \subseteq B_1 \subseteq \ldots \subseteq K \subseteq B_r$.

$$Vol(K) = \frac{Vol(K \cap B_r)}{Vol(K \cap B_{r-1})} \cdot \frac{Vol(K \cap B_{r-1})}{Vol(K \cap B_{r-2})} \cdot \ldots \cdot \frac{Vol(K \cap B_1)}{Vol(K \cap B_0)} \cdot Vol(K \cap B_0)$$

$Vol(K \cap B_0) = Vol(B_0)$ known.

Estimate each ratio $\frac{Vol(K \cap B_i)}{Vol(K \cap B_{i-1})}$.
Sample uniformly at random from $K \cap B_i$ using MCMC and count the proportion of samples falling into $B_{i-1}$.

To ensure that the number of samples needed is small, ensure that the ratio $\frac{Vol(K \cap B_i)}{Vol(K \cap B_{i-1})}$ is small by making the balls grow slowly.
This implies $r = cn \log n$ for some constant $c$.

# Time complexity

The original algorithm has time complexity $O(n^{23})$.

Later it was improved to $O(n^4)$.

Key ingredient: sample uniformly at random from from the points in a convex body in polynomial time. For this, the Markov chain has to converge in polynomial time to the uniform distribution.

# The random walk on cubes

1. Divide the space into $n$-dimensional (hyper)cubes of side $\delta$.

   Choose $\delta$ such to provide a good approximation of $K$, while permitting the random walk on the Markov chain to converge to the stationary distribution in reasonable time.
2. Perform a random walk as follows. If $C$ is the cube at time $t$, select uniformly at random an orthogonally adjacent cube $C'$.

   If $C'$ is in $K$, then move to $C'$, otherwise stay at $C$.

Properties:

- The uniform distribution is the unique stationary distribution.
- **Rapid mixing:** The Markov chain converges to the stationary distribution in number of steps polynomial in $n$.

# A ball walk

Lovász and Simonovits proposed a walk with continuous space.

1. Pick $\delta \in \mathbb{R}$ by the same criteria as before.
2. Perform a random walk as follows.

   If at time $t$ the walk is at $x \in \mathbb{R}^n$, the probability density function at time $t+1$ is uniform over $K \cap B(x, \delta)$ and $0$ outside.

Properties:

- Rapid mixing argument similar to the walk on cubes.
- Saves a factor $n$ in the number of oracle calls.
- Moves more complex, so no saving in time complexity.

# Conclusion

**Theorem**

If we can sample almost uniformly at random from $\Omega_{\text{KNAPSACK}}$ in polynomial time, then there is a polynomial-time randomized approximation scheme for the knapsack counting problem.

**Theorem**

There exists a polynomial time randomized approximation scheme for the volume computation problem.

# Summary of today's lecture

- Randomized algorithms:
  Power of randomness in computation, complexity classes $\mathbf{RP}$ and $\mathbf{BPP}$, error reduction

- Monte Carlo methods:
  Estimating the probability of a random event,
  **model counting via random sampling**

- Markov chain Monte Carlo methods:
  Markov chains and random walks, sampling via MCMC,
  **model counting via MCMC**

# Agenda

| | |
|---|---|
| **Tuesday** | computational complexity, probability theory |
| **Wednesday** | randomized algorithms, Monte Carlo methods |
| **Thursday** | hashing-based approach to model counting |
| **Friday** | from discrete to continuous model counting |