Monte Carlo Method

Vamshi Krishna Vudepu[1]

[1] Lane Department of Computer Science and Electrical Engineering
West Virginia University

05 April, 2012

Outline

Outline

Outline

Definition

What is a Monte Carlo Method?

Definition

What is a Monte Carlo Method?

The Monte Carlo Method refers to a collection of tools for evaluating values through sampling and simulation.

Definition

What is a Monte Carlo Method?

The Monte Carlo Method refers to a collection of tools for evaluating values through sampling and simulation.

Why Monte Carlo Method?

Definition

What is a Monte Carlo Method?

The Monte Carlo Method refers to a collection of tools for evaluating values through sampling and simulation.

Why Monte Carlo Method?

(i) Used when inputs are uncertain and are specified as probability distribution.

Definition

What is a Monte Carlo Method?

The Monte Carlo Method refers to a collection of tools for evaluating values through sampling and simulation.

Why Monte Carlo Method?

(i) Used when inputs are uncertain and are specified as probability distribution.

(ii) Predicts output values based on input samples.

Definition

General steps involved in Monte Carlo Methods

Definition

General steps involved in Monte Carlo Methods

(i) Define a domain of possible inputs.

Definition

General steps involved in Monte Carlo Methods

(i) Define a domain of possible inputs.

(ii) Choose inputs at random from the probability distribution over the domain.

Definition

General steps involved in Monte Carlo Methods

(i) Define a domain of possible inputs.
(ii) Choose inputs at random from the probability distribution over the domain.
(iii) Perform computation on inputs and aggregate the results.

Definition

Sampling

Definition

Sampling

How can you ensure that an input is selected uniformly at random from the given input domain in Computer Simulations?

Definition

Sampling

How can you ensure that an input is selected uniformly at random from the given input domain in Computer Simulations? Using Random number generators.

Definition

Sampling

How can you ensure that an input is selected uniformly at random from the given input domain in Computer Simulations? Using Random number generators. Consider 3 black balls and 7 red balls in a bin.

Definition

Sampling

How can you ensure that an input is selected uniformly at random from the given input domain in Computer Simulations? Using Random number generators. Consider 3 black balls and 7 red balls in a bin. The probability of choosing a black ball and a red ball is is 0.3 and 0.7 respectively.

Definition

Sampling

How can you ensure that an input is selected uniformly at random from the given input domain in Computer Simulations? Using Random number generators. Consider 3 black balls and 7 red balls in a bin. The probability of choosing a black ball and a red ball is is 0.3 and 0.7 respectively. A random number generator which generates any number between 0 and 1 (equally likely) can be used to choose a ball.

Definition

Sampling

How can you ensure that an input is selected uniformly at random from the given input domain in Computer Simulations? Using Random number generators. Consider 3 black balls and 7 red balls in a bin. The probability of choosing a black ball and a red ball is is 0.3 and 0.7 respectively. A random number generator which generates any number between 0 and 1 (equally likely) can be used to choose a ball. We can associate each possible random number generated to the input.

Definition

Sampling

How can you ensure that an input is selected uniformly at random from the given input domain in Computer Simulations? Using Random number generators. Consider 3 black balls and 7 red balls in a bin. The probability of choosing a black ball and a red ball is is 0.3 and 0.7 respectively. A random number generator which generates any number between 0 and 1 (equally likely) can be used to choose a ball. We can associate each possible random number generated to the input. If the number generated is less than or equal to 0.3, then a black ball will be picked.

Definition

Sampling

How can you ensure that an input is selected uniformly at random from the given input domain in Computer Simulations? Using Random number generators. Consider 3 black balls and 7 red balls in a bin. The probability of choosing a black ball and a red ball is is 0.3 and 0.7 respectively. A random number generator which generates any number between 0 and 1 (equally likely) can be used to choose a ball. We can associate each possible random number generated to the input. If the number generated is less than or equal to 0.3, then a black ball will be picked. For the numbers between 0.3 and 1, red ball will be selected.

Outline

Examples

Example

Examples

Example

Consider two fair dice.Calculate the probability for particular sum of outcomes of the two dice.

Examples

Example

Consider two fair dice.Calculate the probability for particular sum of outcomes of the two dice. For example, the probability that the sum shows up 6 is 0.138.

Examples

Example

Consider two fair dice.Calculate the probability for particular sum of outcomes of the two dice. For example, the probability that the sum shows up 6 is 0.138. Roll the dice manually 100 times and note down how many times the sum six turns up on the dice.

Examples

Example

Consider two fair dice.Calculate the probability for particular sum of outcomes of the two dice. For example, the probability that the sum shows up 6 is 0.138. Roll the dice manually 100 times and note down how many times the sum six turns up on the dice. If the sum six showed up 15 times, the probability could be concluded as 0.15.

Examples

Example

Consider two fair dice.Calculate the probability for particular sum of outcomes of the two dice. For example, the probability that the sum shows up 6 is 0.138. Roll the dice manually 100 times and note down how many times the sum six turns up on the dice. If the sum six showed up 15 times, the probability could be concluded as 0.15. How can you increase accuracy?

Examples

Example

Consider two fair dice.Calculate the probability for particular sum of outcomes of the two dice. For example, the probability that the sum shows up 6 is 0.138. Roll the dice manually 100 times and note down how many times the sum six turns up on the dice. If the sum six showed up 15 times, the probability could be concluded as 0.15. How can you increase accuracy? Accuracy increases with increase in number of times we roll the dice.

Examples

Example

Consider two fair dice.Calculate the probability for particular sum of outcomes of the two dice. For example, the probability that the sum shows up 6 is 0.138. Roll the dice manually 100 times and note down how many times the sum six turns up on the dice. If the sum six showed up 15 times, the probability could be concluded as 0.15. How can you increase accuracy? Accuracy increases with increase in number of times we roll the dice.

Monte Carlo method simulates rolling of dice for given number of times (say for instance 1000 times) and gives the probability.

Examples

Example

Examples

Example

Estimate the value of π

Examples

Example

Estimate the value of π
Draw a square of 2×2 units on floor and inscribe a circle in it.

Examples

Example

Estimate the value of π
Draw a square of 2×2 units on floor and inscribe a circle in it. Scatter some objects of same size (such as grains) at random.

Examples

Example

Estimate the value of π
Draw a square of 2×2 units on floor and inscribe a circle in it. Scatter some objects of same size (such as grains) at random. What is the probability that the object falls inside the circle?

Examples

Example

Estimate the value of π

Draw a square of 2×2 units on floor and inscribe a circle in it. Scatter some objects of same size (such as grains) at random. What is the probability that the object falls inside the circle? Ratio of area of circle to the area of square

Examples

Example

Estimate the value of π

Draw a square of 2×2 units on floor and inscribe a circle in it. Scatter some objects of same size (such as grains) at random. What is the probability that the object falls inside the circle? Ratio of area of circle to the area of square which is

Examples

Example

Estimate the value of π
Draw a square of 2×2 units on floor and inscribe a circle in it. Scatter some objects of same size (such as grains) at random. What is the probability that the object falls inside the circle? Ratio of area of circle to the area of square which is $\frac{\pi}{4}$.

Examples

Example

Estimate the value of π
Draw a square of 2×2 units on floor and inscribe a circle in it. Scatter some objects of same size (such as grains) at random. What is the probability that the object falls inside the circle? Ratio of area of circle to the area of square which is $\frac{\pi}{4}$. Ratio of number of grains inside the circle to the total number of grains inside the square gives approximately the same value.

Examples

Example

Estimate the value of π

Draw a square of 2×2 units on floor and inscribe a circle in it. Scatter some objects of same size (such as grains) at random. What is the probability that the object falls inside the circle? Ratio of area of circle to the area of square which is $\frac{\pi}{4}$. Ratio of number of grains inside the circle to the total number of grains inside the square gives approximately the same value. Multiplying with 4 gives the approximate value of π.

Examples

Example

Estimate the value of π
Draw a square of 2×2 units on floor and inscribe a circle in it. Scatter some objects of same size (such as grains) at random. What is the probability that the object falls inside the circle? Ratio of area of circle to the area of square which is $\frac{\pi}{4}$. Ratio of number of grains inside the circle to the total number of grains inside the square gives approximately the same value. Multiplying with 4 gives the approximate value of π.

Note

Examples

Example

Estimate the value of π
Draw a square of 2×2 units on floor and inscribe a circle in it. Scatter some objects of same size (such as grains) at random. What is the probability that the object falls inside the circle? Ratio of area of circle to the area of square which is $\frac{\pi}{4}$. Ratio of number of grains inside the circle to the total number of grains inside the square gives approximately the same value. Multiplying with 4 gives the approximate value of π.

Note

To get the accurate value

Examples

Example

Estimate the value of π
Draw a square of 2×2 units on floor and inscribe a circle in it. Scatter some objects of same size (such as grains) at random. What is the probability that the object falls inside the circle? Ratio of area of circle to the area of square which is $\frac{\pi}{4}$. Ratio of number of grains inside the circle to the total number of grains inside the square gives approximately the same value. Multiplying with 4 gives the approximate value of π.

Note

To get the accurate value

(i) *The number of objects should be large enough.*

Examples

Example

Estimate the value of π
Draw a square of 2×2 units on floor and inscribe a circle in it. Scatter some objects of same size (such as grains) at random. What is the probability that the object falls inside the circle? Ratio of area of circle to the area of square which is $\frac{\pi}{4}$. Ratio of number of grains inside the circle to the total number of grains inside the square gives approximately the same value. Multiplying with 4 gives the approximate value of π.

Note

To get the accurate value

(i) *The number of objects should be large enough.*

(ii) *Each object has to be scattered uniformly at random. Object should not be dropped purposefully at a particular location inside the square.*

Examples

Example

Examples

Example

Estimating π value using Monte Carlo method

Examples

Example

Estimating π value using Monte Carlo method
Let (X, Y) be a point chosen uniformly at random in a 2×2 square centered at origin $(0, 0)$.

Examples

Example

Estimating π value using Monte Carlo method
Let (X, Y) be a point chosen uniformly at random in a 2×2 square centered at origin $(0, 0)$. Consider a circle of radius 1 unit centered at origin inscribed in the square.

Examples

Example

Estimating π value using Monte Carlo method
Let (X, Y) be a point chosen uniformly at random in a 2×2 square centered at origin $(0, 0)$. Consider a circle of radius 1 unit centered at origin inscribed in the square. Consider a random variable Z such that

$$Z = \left\{ \right.$$

Examples

Example

Estimating π value using Monte Carlo method
Let (X, Y) be a point chosen uniformly at random in a 2×2 square centered at origin $(0, 0)$. Consider a circle of radius 1 unit centered at origin inscribed in the square.
Consider a random variable Z such that

$$Z = \begin{cases} 1, & \text{if } (X, Y) \text{ lies inside the circle,} \\ \end{cases}$$

Examples

Example

Estimating π value using Monte Carlo method
Let (X, Y) be a point chosen uniformly at random in a 2×2 square centered at origin $(0, 0)$. Consider a circle of radius 1 unit centered at origin inscribed in the square.
Consider a random variable Z such that

$$Z = \begin{cases} 1, & \text{if } (X, Y) \text{ lies inside the circle,} \\ 0, & \text{Otherwise.} \end{cases}$$

Examples

Example

Estimating π value using Monte Carlo method

Let (X, Y) be a point chosen uniformly at random in a 2×2 square centered at origin $(0, 0)$. Consider a circle of radius 1 unit centered at origin inscribed in the square.
Consider a random variable Z such that

$$Z = \begin{cases} 1, & \text{if } (X, Y) \text{ lies inside the circle,} \\ 0, & \text{Otherwise.} \end{cases}$$

$$Pr(Z = 1) = \frac{\pi}{4}$$

Examples

Example

Estimating π value using Monte Carlo method
Let (X, Y) be a point chosen uniformly at random in a 2×2 square centered at origin $(0, 0)$. Consider a circle of radius 1 unit centered at origin inscribed in the square. Consider a random variable Z such that

$$Z = \begin{cases} 1, & \text{if } (X, Y) \text{ lies inside the circle,} \\ 0, & \text{Otherwise.} \end{cases}$$

$$Pr(Z = 1) = \frac{\pi}{4}$$

Assuming that the experiment is run m times with X and Y chosen independently among the runs, let

$$W = \sum_{i=1}^{m} Z_i$$

Examples

Example

Estimating π value using Monte Carlo method
Let (X, Y) be a point chosen uniformly at random in a 2×2 square centered at origin $(0, 0)$. Consider a circle of radius 1 unit centered at origin inscribed in the square.
Consider a random variable Z such that

$$Z = \begin{cases} 1, & \text{if } (X, Y) \text{ lies inside the circle,} \\ 0, & \text{Otherwise.} \end{cases}$$

$$Pr(Z = 1) = \frac{\pi}{4}$$

Assuming that the experiment is run m times with X and Y chosen independently among the runs, let

$$W = \sum_{i=1}^{m} Z_i$$

$$\mathbf{E}[W] = \mathbf{E}\left[\sum_{i=1}^{m} Z_i\right] = \sum_{i=1}^{m} \mathbf{E}[Z_i] = \frac{m \cdot \pi}{4}$$

Examples

Example

Examples

Example

Hence, $W' = \frac{4}{m} \cdot W$ is an estimate of π.

Examples

Example

Hence, $W' = \frac{4}{m} \cdot W$ is an estimate of π.

Applying the following Chernoff bound,

Examples

Example

Hence, $W' = \frac{4}{m} \cdot W$ is an estimate of π.

Applying the following Chernoff bound,

$$Pr(\ |X - \mu\ | \geq \delta \cdot \mu) \quad \leq \quad 2 \cdot e^{-\mu \cdot \delta^2/3}$$

Examples

Example

Hence, $W' = \frac{4}{m} \cdot W$ is an estimate of π.

Applying the following Chernoff bound,

$$\begin{aligned} Pr(\,|X - \mu\,| \geq \delta \cdot \mu) &\leq 2 \cdot e^{-\mu \cdot \delta^2/3} \\ Pr(\,|W' - \Pi\,| \geq \epsilon \cdot \pi) &= Pr(\,|W - \frac{m \cdot \pi|}{4}\,| \geq \frac{\epsilon \cdot m \cdot \pi}{4}) \end{aligned}$$

Examples

Example

Hence, $W' = \frac{4}{m} \cdot W$ is an estimate of π.

Applying the following Chernoff bound,

$$\begin{aligned} Pr(\,|X - \mu\,| \geq \delta \cdot \mu) &\leq 2 \cdot e^{-\mu \cdot \delta^2/3} \\ Pr(\,|W' - \Pi\,| \geq \epsilon \cdot \pi) &= Pr(\,|W - \frac{m \cdot \pi|}{4}\,| \geq \frac{\epsilon \cdot m \cdot \pi}{4}) \\ &= Pr(\,|W - \mathbf{E}[W]\,| \geq \epsilon \cdot \mathbf{E}[W]) \end{aligned}$$

Examples

Example

Hence, $W' = \frac{4}{m} \cdot W$ is an estimate of π.

Applying the following Chernoff bound,

$$\begin{aligned}
Pr(\,|X - \mu\,| \geq \delta \cdot \mu) &\leq 2 \cdot e^{-\mu \cdot \delta^2/3} \\
Pr(\,|W' - \Pi\,| \geq \epsilon \cdot \pi) &= Pr(\,|W - \frac{m \cdot \pi|}{4}\,| \geq \frac{\epsilon \cdot m \cdot \pi}{4}) \\
&= Pr(\,|W - \mathbf{E}[W]\,| \geq \epsilon \cdot \mathbf{E}[W]) \\
&\leq 2 \cdot e^{-m \cdot \Pi \cdot \epsilon^2/12}
\end{aligned}$$

Examples

Example

Hence, $W' = \frac{4}{m} \cdot W$ is an estimate of π.

Applying the following Chernoff bound,

$$\begin{aligned} Pr(\ |X - \mu\ | \geq \delta \cdot \mu) &\leq 2 \cdot e^{-\mu \cdot \delta^2/3} \\ Pr(\ |W' - \Pi\ | \geq \epsilon \cdot \pi) &= Pr(\ |W - \frac{m \cdot \pi|}{4}\ | \geq \frac{\epsilon \cdot m \cdot \pi}{4}) \\ &= Pr(\ |W - \mathbf{E}[W]\ | \geq \epsilon \cdot \mathbf{E}[W]) \\ &\leq 2 \cdot e^{-m \cdot \Pi \cdot \epsilon^2/12} \end{aligned}$$

Therefore, m should be sufficiently large to obtain a tight approximation of π w.h.p.

Outline

1. Introduction
   - Monte Carlo Definition
   - Examples
   - Definitions
2. Applications
   - The DNF Counting Problem
   - DNF Couting Algorithms

Definitions

Definition

Definitions

Definition

A randomized algorithm gives an (ϵ, δ)- approximation for the value V if the output X of the algorithm satisfies

Definitions

Definition

A randomized algorithm gives an (ϵ, δ)- approximation for the value V if the output X of the algorithm satisfies

$$Pr(\ |X - V\ | \leq \epsilon V) \geq 1 - \delta$$

Definitions

Definition

A randomized algorithm gives an (ϵ, δ)- approximation for the value V if the output X of the algorithm satisfies

$$Pr(\ |X - V\ | \leq \epsilon V) \geq 1 - \delta$$

Question

Definitions

Definition

A randomized algorithm gives an (ϵ, δ)- approximation for the value V if the output X of the algorithm satisfies

$$Pr(\ |X - V\ | \leq \epsilon V) \geq 1 - \delta$$

Question

In the π example, what should be the value of m for the algorithm to give an (ϵ, δ)-appromixation?

Definitions

Definition

A randomized algorithm gives an (ϵ, δ)- approximation for the value V if the output X of the algorithm satisfies

$$Pr(|X - V| \leq \epsilon V) \geq 1 - \delta$$

Question

In the π example, what should be the value of m for the algorithm to give an (ϵ, δ)-appromixation?
As long as $\epsilon \leq 1$,

Definitions

Definition

A randomized algorithm gives an (ϵ, δ)- approximation for the value V if the output X of the algorithm satisfies

$$Pr(\ |X - V\ | \leq \epsilon V) \geq 1 - \delta$$

Question

In the π example, what should be the value of m for the algorithm to give an (ϵ, δ)-appromixation?
As long as $\epsilon \leq 1$, m should be large enough to make

$$2 \cdot e^{\frac{-m \cdot \pi \cdot \epsilon^2}{12}} \leq \delta$$

Definitions

Definition

A randomized algorithm gives an (ϵ, δ)- approximation for the value V if the output X of the algorithm satisfies

$$Pr(\ |X - V\ | \leq \epsilon V) \geq 1 - \delta$$

Question

In the π example, what should be the value of m for the algorithm to give an (ϵ, δ)-appromixation?
As long as $\epsilon \leq 1$, m should be large enough to make

$$2 \cdot e^{\frac{-m \cdot \pi \cdot \epsilon^2}{12}} \leq \delta$$

Solving the above equation, we get

$$m \geq \frac{12 \cdot \ln(2/\delta)}{\pi \cdot \epsilon^2}$$

Definitions

Theorem

Definitions

Theorem

Let $X_1, ..., X_m$ be independent and identically distributed indicator random variables,

Definitions

Theorem

Let $X_1, ..., X_m$ be independent and identically distributed indicator random variables, with $\mu = \mathbf{E}[X_i]$.

Definitions

Theorem

Let $X_1, \ldots, X_m$ be independent and identically distributed indicator random variables, with $\mu = \mathbf{E}[X_i]$. If $m \geq \frac{3 \cdot \ln(2/\delta)}{\epsilon^2 \cdot \mu}$, then

Definitions

Theorem

Let $X_1, ..., X_m$ be independent and identically distributed indicator random variables, with $\mu = \mathbf{E}[X_i]$. If $m \geq \frac{3 \cdot \ln(2/\delta)}{\epsilon^2 \cdot \mu}$, then

$$Pr\left(\left|\frac{1}{m}\sum_{i=1}^{m} X_i - \mu\right| \geq \epsilon \cdot \mu\right) \leq \delta$$

Definitions

Theorem

Let $X_1, ..., X_m$ be independent and identically distributed indicator random variables, with $\mu = \mathbf{E}[X_i]$. If $m \geq \frac{3 \cdot \ln(2/\delta)}{\epsilon^2 \cdot \mu}$, then

$$Pr\left(\left|\frac{1}{m}\sum_{i=1}^{m} X_i - \mu\right| \geq \epsilon \cdot \mu\right) \leq \delta$$

i.e, m samples provides an (ϵ, δ) approximation for μ.

Definitions

Definition

Definitions

Definition

A fully polynomial randomized approximation scheme (FPRAS) for a problem is a randomized algorithm for which,

Definitions

Definition

A fully polynomial randomized approximation scheme (FPRAS) for a problem is a randomized algorithm for which, given an input x and any parameters ϵ and δ with $0 < \epsilon, \delta < 1$,

Definitions

Definition

A fully polynomial randomized approximation scheme (FPRAS) for a problem is a randomized algorithm for which, given an input x and any parameters ϵ and δ with $0 < \epsilon, \delta < 1$, the algorithm outputs an (ϵ, δ)-approximation to $V(x)$ in time that is polynomial in

Definitions

Definition

A fully polynomial randomized approximation scheme (FPRAS) for a problem is a randomized algorithm for which, given an input x and any parameters ϵ and δ with $0 < \epsilon, \delta < 1$, the algorithm outputs an (ϵ, δ)-approximation to $V(x)$ in time that is polynomial in $1/\epsilon$,

Definitions

Definition

A fully polynomial randomized approximation scheme (FPRAS) for a problem is a randomized algorithm for which, given an input x and any parameters ϵ and δ with $0 < \epsilon, \delta < 1$, the algorithm outputs an (ϵ, δ)-approximation to $V(x)$ in time that is polynomial in $1/\epsilon$, $\ln \delta^{-1}$,

Definitions

Definition

A fully polynomial randomized approximation scheme (FPRAS) for a problem is a randomized algorithm for which, given an input x and any parameters ϵ and δ with $0 < \epsilon, \delta < 1$, the algorithm outputs an (ϵ, δ)-approximation to $V(x)$ in time that is polynomial in $1/\epsilon$, $\ln \delta^{-1}$, and the size of the input x.

Outline

DNF

Note

DNF

Note

Sampling in Monte Carlo methods is not as simple as shown for estimating Π value.

DNF

Note

Sampling in Monte Carlo methods is not as simple as shown for estimating Π value. Sampling is often nontrivial.

DNF

Note

Sampling in Monte Carlo methods is not as simple as shown for estimating Π value. Sampling is often nontrivial.

What is a DNF?

DNF

Note

Sampling in Monte Carlo methods is not as simple as shown for estimating Π *value. Sampling is often nontrivial.*

What is a DNF?

A Boolean formula which comprises of disjuntion (OR) of clauses where each clause is a conjunction (AND) of literals.

DNF

Note

Sampling in Monte Carlo methods is not as simple as shown for estimating Π value. Sampling is often nontrivial.

What is a DNF?

A Boolean formula which comprises of disjuntion (OR) of clauses where each clause is a conjunction (AND) of literals.

Example

DNF

Note

Sampling in Monte Carlo methods is not as simple as shown for estimating Π value. Sampling is often nontrivial.

What is a DNF?

A Boolean formula which comprises of disjuntion (OR) of clauses where each clause is a conjunction (AND) of literals.

Example

$(x_1 \wedge \bar{x}_2) \vee (x_3 \wedge x_4) \vee (\bar{x}_4 \wedge x_1)$

DNF

Questions

DNF

Questions

(i) What is a satisfying assignment?

DNF

Questions

(i) What is a satisfying assignment? An assignment which when given to a Boolean formula returns TRUE.

DNF

Questions

(i) What is a satisfying assignment? An assignment which when given to a Boolean formula returns TRUE.

(ii) Determining the satisfiability of CNF is simpler compared to DNF. True/False?

DNF

Questions

(i) What is a satisfying assignment? An assignment which when given to a Boolean formula returns TRUE.

(ii) Determining the satisfiability of CNF is simpler compared to DNF. True/False? False.

DNF

Questions

(i) What is a satisfying assignment? An assignment which when given to a Boolean formula returns TRUE.

(ii) Determining the satisfiability of CNF is simpler compared to DNF. True/False? False. The input assignment needs to satisfy just one clause in a DNF.

DNF

Questions

(i) What is a satisfying assignment? An assignment which when given to a Boolean formula returns TRUE.

(ii) Determining the satisfiability of CNF is simpler compared to DNF. True/False? False. The input assignment needs to satisfy just one clause in a DNF.

(iii) How to check the satisfiability of a CNF?

DNF

Questions

(i) What is a satisfying assignment? An assignment which when given to a Boolean formula returns TRUE.

(ii) Determining the satisfiability of CNF is simpler compared to DNF. True/False? False. The input assignment needs to satisfy just one clause in a DNF.

(iii) How to check the satisfiability of a CNF? Convert it into DNF using

DNF

Questions

(i) What is a satisfying assignment? An assignment which when given to a Boolean formula returns TRUE.

(ii) Determining the satisfiability of CNF is simpler compared to DNF. True/False? False. The input assignment needs to satisfy just one clause in a DNF.

(iii) How to check the satisfiability of a CNF? Convert it into DNF using De Morgan's law.

DNF

Questions

(i) What is a satisfying assignment? An assignment which when given to a Boolean formula returns TRUE.

(ii) Determining the satisfiability of CNF is simpler compared to DNF. True/False? False. The input assignment needs to satisfy just one clause in a DNF.

(iii) How to check the satisfiability of a CNF? Convert it into DNF using De Morgan's law.

(iv) If H is a CNF of n Boolean variables and $\bar{H}$ is its DNF, what is the maximum number of satisfying assignments can $\bar{H}$ have for H to have a satisfying assignment?

DNF

Questions

(i) What is a satisfying assignment? An assignment which when given to a Boolean formula returns TRUE.

(ii) Determining the satisfiability of CNF is simpler compared to DNF. True/False? False. The input assignment needs to satisfy just one clause in a DNF.

(iii) How to check the satisfiability of a CNF? Convert it into DNF using De Morgan's law.

(iv) If H is a CNF of n Boolean variables and $\bar{H}$ is its DNF, what is the maximum number of satisfying assignments can $\bar{H}$ have for H to have a satisfying assignment? Strictly less than 2^n.

DNF

Questions

(i) What is a satisfying assignment? An assignment which when given to a Boolean formula returns TRUE.

(ii) Determining the satisfiability of CNF is simpler compared to DNF. True/False? False. The input assignment needs to satisfy just one clause in a DNF.

(iii) How to check the satisfiability of a CNF? Convert it into DNF using De Morgan's law.

(iv) If H is a CNF of n Boolean variables and $\bar{H}$ is its DNF, what is the maximum number of satisfying assignments can $\bar{H}$ have for H to have a satisfying assignment? Strictly less than 2^n.

Note

DNF

Questions

(i) What is a satisfying assignment? An assignment which when given to a Boolean formula returns TRUE.

(ii) Determining the satisfiability of CNF is simpler compared to DNF. True/False? False. The input assignment needs to satisfy just one clause in a DNF.

(iii) How to check the satisfiability of a CNF? Convert it into DNF using De Morgan's law.

(iv) If H is a CNF of n Boolean variables and $\bar{H}$ is its DNF, what is the maximum number of satisfying assignments can $\bar{H}$ have for H to have a satisfying assignment? Strictly less than 2^n.

Note

Counting the number of satisfying assignments for a DNF comes under $\#P-$ *Complete Problems.*

Outline

1. Introduction
   - Monte Carlo Definition
   - Examples
   - Definitions
2. Applications
   - The DNF Counting Problem
   - DNF Couting Algorithms

DNF Counting Algorithm

Input: A DNF formula F with n variables.
Output: $Y =$ an approximation of $c(F)$.

1: $X \leftarrow 0$
2: **for** $(k = 1 \text{ to } m)$ **do**
3: Generate a random assignment for n variables, chosen uniformly at random from all 2^n possible assignments.
4: **if** (the random assignment satisfies F) **then**
5: $X \leftarrow X + 1$
6: **end if**
7: **end for**
8: **return** $Y \leftarrow (\frac{X}{m}) \cdot 2^n$

Algorithm 3.1: DNF Counting Algorithm I: Naive approach

DNF- Counting Algorithm

DNF- Counting Algorithm

Notes

$c(F) \rightarrow$ No. of satisfying assignments of DNF Formaula F.

DNF- Counting Algorithm

Notes

$c(F) \rightarrow$ No. of satisfying assignments of DNF Formaula F.
Let X_k be random variable such that

$$X_k = \left\{ \right.$$

DNF- Counting Algorithm

Notes

$c(F) \rightarrow$ No. of satisfying assignments of DNF Formaula F.
Let X_k be random variable such that

$$X_k = \begin{cases} 1, & \text{if } k^{\text{th}} \text{ iteration generates a satisfying assignment.} \end{cases}$$

DNF- Counting Algorithm

Notes

$c(F) \rightarrow$ No. of satisfying assignments of DNF Formaula F.
Let X_k be random variable such that

$$X_k = \begin{cases} 1, & \text{if } k^{\text{th}} \text{ iteration generates a satisfying assignment.} \\ 0, & \text{Otherwise.} \end{cases}$$

DNF- Counting Algorithm

Notes

$c(F) \rightarrow$ No. of satisfying assignments of DNF Formaula F.
Let X_k be random variable such that

$$X_k = \begin{cases} 1, & \text{if } k^{\text{th}} \text{ iteration generates a satisfying assignment.} \\ 0, & \text{Otherwise.} \end{cases}$$

$$X = \sum_{k=1}^{m} X_k.$$

DNF- Counting Algorithm

Notes

$c(F) \rightarrow$ No. of satisfying assignments of DNF Formaula F.
Let X_k be random variable such that

$$X_k = \begin{cases} 1, & \text{if } k^{\text{th}} \text{ iteration generates a satisfying assignment.} \\ 0, & \text{Otherwise.} \end{cases}$$

$$X = \sum_{k=1}^{m} X_k.$$

What is the probability that X_k is 1?

DNF- Counting Algorithm

Notes

$c(F) \to$ No. of satisfying assignments of DNF Formaula F.
Let X_k be random variable such that

$$X_k = \begin{cases} 1, & \text{if } k^{\text{th}} \text{ iteration generates a satisfying assignment.} \\ 0, & \text{Otherwise.} \end{cases}$$

$$X = \sum_{k=1}^{m} X_k.$$

What is the probability that X_k is 1? $\frac{c(F)}{2^n}$

DNF- Counting Algorithm

Notes

$c(F) \rightarrow$ No. of satisfying assignments of DNF Formaula F.
Let X_k be random variable such that

$$X_k = \begin{cases} 1, & \text{if } k^{\text{th}} \text{ iteration generates a satisfying assignment.} \\ 0, & \text{Otherwise.} \end{cases}$$

$$X = \sum_{k=1}^{m} X_k.$$

What is the probability that X_k is 1? $\frac{c(F)}{2^n}$

$$\mathbf{E}[Y] = \frac{\mathbf{E}[X] \cdot 2^n}{m} = c(F)$$

DNF- Counting Algorithm

Notes

$c(F) \rightarrow$ No. of satisfying assignments of DNF Formaula F.
Let X_k be random variable such that

$$X_k = \begin{cases} 1, & \text{if } k^{\text{th}} \text{ iteration generates a satisfying assignment.} \\ 0, & \text{Otherwise.} \end{cases}$$

$$X = \sum_{k=1}^{m} X_k.$$

What is the probability that X_k is 1? $\frac{c(F)}{2^n}$

$$\mathbf{E}[Y] = \frac{\mathbf{E}[X] \cdot 2^n}{m} = c(F)$$

Applying our previous theorem, Y gives an (ϵ, δ) -approximation of $c(F)$, when

DNF- Counting Algorithm

Notes

$c(F) \rightarrow$ No. of satisfying assignments of DNF Formaula F.
Let X_k be random variable such that

$$X_k = \begin{cases} 1, & \text{if } k^{\text{th}} \text{ iteration generates a satisfying assignment.} \\ 0, & \text{Otherwise.} \end{cases}$$

$$X = \sum_{k=1}^{m} X_k.$$

What is the probability that X_k is 1? $\frac{c(F)}{2^n}$

$$\mathbf{E}[Y] = \frac{\mathbf{E}[X] \cdot 2^n}{m} = c(F)$$

Applying our previous theorem, Y gives an (ϵ, δ) -approximation of $c(F)$, when

$$m \geq \frac{3 \cdot 2^n \cdot \ln(2/\delta)}{\epsilon^2 \cdot c(F)}$$

DNF-Counting Algorithm

Problems in Naive Approach

DNF-Counting Algorithm

Problems in Naive Approach

If $c(F) \geq 2^n/\alpha(n)$ for some polynomial α, then we need number of samples m polynomial in n, $1/\epsilon$, $\ln(1/\delta)$.

DNF-Counting Algorithm

Problems in Naive Approach

If $c(F) \geq 2^n/\alpha(n)$ for some polynomial α, then we need number of samples m polynomial in n, $1/\epsilon$, $\ln(1/\delta)$.

However, $c(F)$ can be much less than 2^n.

DNF-Counting Algorithm

Problems in Naive Approach

If $c(F) \geq 2^n/\alpha(n)$ for some polynomial α, then we need number of samples m polynomial in n, $1/\epsilon$, $\ln(1/\delta)$.

However, $c(F)$ can be much less than 2^n. In such case, it takes exponential number of assignments before finding the first satisfying assignment.

DNF-Counting Algorithm

Problems in Naive Approach

If $c(F) \geq 2^n/\alpha(n)$ for some polynomial α, then we need number of samples m polynomial in n, $1/\epsilon$, $\ln(1/\delta)$.

However, $c(F)$ can be much less than 2^n. In such case, it takes exponential number of assignments before finding the first satisfying assignment.

The number of satisfying assignments might not be sufficiently dense enough in set of all assignments.

DNF-Counting Algorithm

Problems in Naive Approach

If $c(F) \geq 2^n/\alpha(n)$ for some polynomial α, then we need number of samples m polynomial in n, $1/\epsilon$, $\ln(1/\delta)$.

However, $c(F)$ can be much less than 2^n. In such case, it takes exponential number of assignments before finding the first satisfying assignment.

The number of satisfying assignments might not be sufficiently dense enough in set of all assignments.

To obtain FPRAS for this problem, we need to construct a better sample space that includes all the satisfying assignments of F and also, the satisfying assignments must be sufficiently dense in the sample space.

DNF- Counting Algorithm

FPRAS for DNF Couting

DNF- Counting Algorithm

FPRAS for DNF Couting

Let $F = C_1 \vee C_2 \vee \cdots \vee C_t$

DNF- Counting Algorithm

FPRAS for DNF Couting

$$\text{Let } F = C_1 \vee C_2 \vee \cdots \vee C_t$$

A satisfying assignment needs to satisfy at least one clause.

DNF- Counting Algorithm

FPRAS for DNF Couting

$$\text{Let } F = C_1 \vee C_2 \vee \cdots \vee C_t$$

A satisfying assignment needs to satisfy at least one clause.
Let SC_i be set of assignments that satisfy clause i

DNF- Counting Algorithm

FPRAS for DNF Couting

$$\text{Let } F = C_1 \vee C_2 \vee \cdots \vee C_t$$

A satisfying assignment needs to satisfy at least one clause.
Let SC_i be set of assignments that satisfy clause i

$$\text{Let } U = \{(i, a) \mid 1 \leq i \leq t \ \ \& \ a \in SC_i\}$$

DNF- Counting Algorithm

FPRAS for DNF Couting

$$\text{Let } F = C_1 \vee C_2 \vee \cdots \vee C_t$$

A satisfying assignment needs to satisfy at least one clause.
Let SC_i be set of assignments that satisfy clause i

$$\text{Let } U = \{(i, a) \mid 1 \leq i \leq t \ \ \& \ a \in SC_i\}$$

U denotes set of satisfying assignments for all clauses.

$$\sum_{i=1}^{t} |SC_i| = |U|$$

DNF-Counting Algorithm

FPRAS for DNF Couting cont...

DNF-Counting Algorithm

FPRAS for DNF Couting cont...

The count can be estimated by

$$c(F) = \left| \bigcup_{i=1}^{t} SC_i \right|$$

DNF-Counting Algorithm

FPRAS for DNF Couting cont...

The count can be estimated by

$$c(F) = \left| \bigcup_{i=1}^{t} SC_i \right|$$

$c(F) \leq |U|$, since an assignment can satisfy more than one clause and thus can appear in more than one pair in U.

DNF-Counting Algorithm

FPRAS for DNF Couting cont...

The count can be estimated by

$$c(F) = \left| \bigcup_{i=1}^{t} SC_i \right|$$

$c(F) \leq |U|$, since an assignment can satisfy more than one clause and thus can appear in more than one pair in U. Let S be subset of U with size $c(F)$.

DNF-Counting Algorithm

FPRAS for DNF Couting cont...

The count can be estimated by

$$c(F) = \left| \bigcup_{i=1}^{t} SC_i \right|$$

$c(F) \leq |U|$, since an assignment can satisfy more than one clause and thus can appear in more than one pair in U. Let S be subset of U with size $c(F)$.

$$S = \{(i, a) \mid 1 \leq i \leq t, a \in SC_i, a \notin SC_j \text{ for } j < i\}$$

DNF-Counting Algorithm

FPRAS for DNF Couting cont...

The count can be estimated by

$$c(F) = \left| \bigcup_{i=1}^{t} SC_i \right|$$

$c(F) \leq |U|$, since an assignment can satisfy more than one clause and thus can appear in more than one pair in U. Let S be subset of U with size $c(F)$.

$$S = \{(i, a) \mid 1 \leq i \leq t, a \in SC_i, a \notin SC_j \text{ for } j < i\}$$

How to sample uniformly from U?

DNF-Counting Algorithm

FPRAS for DNF Couting cont...

The count can be estimated by

$$c(F) = \left| \bigcup_{i=1}^{t} SC_i \right|$$

$c(F) \leq |U|$, since an assignment can satisfy more than one clause and thus can appear in more than one pair in U. Let S be subset of U with size $c(F)$.

$$S = \{(i, a) \mid 1 \leq i \leq t, a \in SC_i, a \notin SC_j \text{ for } j < i\}$$

How to sample uniformly from U? First choose i from the pair (i, a) in U. Since i^{th} clause has $|SC_i|$ satisfying assignments, we should choose i with probability

DNF-Counting Algorithm

FPRAS for DNF Couting cont...

The count can be estimated by

$$c(F) = \left| \bigcup_{i=1}^{t} SC_i \right|$$

$c(F) \leq |U|$, since an assignment can satisfy more than one clause and thus can appear in more than one pair in U. Let S be subset of U with size $c(F)$.

$$S = \{(i, a) \mid 1 \leq i \leq t, a \in SC_i, a \notin SC_j \text{ for } j < i\}$$

How to sample uniformly from U? First choose i from the pair (i, a) in U. Since i^{th} clause has $|SC_i|$ satisfying assignments, we should choose i with probability

$$\frac{|SC_i|}{\sum_{i=1}^{t} |SC_i|} = \frac{|SC_i|}{|U|}$$

DNF-Counting Algorithm

Input: A DNF formula F with n variables.
Output: $Y =$ an approximation of $c(F)$.

$X \leftarrow 0$
for ($k = 1$ to m) **do**
  With probability $\frac{|SC_i|}{\sum_{i=1}^{t} |SC_i|}$ choose, uniformly at random, an assignment $a \in SC_i$.
  if (a is not in any $SC_j, j < i$) **then**
    $X \leftarrow X + 1$
  end if
end for
return $Y \leftarrow (\frac{X}{m}) \cdot \sum_{i=1}^{t} |SC_i|$

Algorithm 3.2: DNF Counting Algorithm II: FPRAS

DNF Counting Algorithm

Theorem

DNF counting algorithm II is a fully polynomial randomized approximation scheme (FPRAS) for the DNF counting problem when

$$m = \lceil (3 \cdot t^2/\epsilon^2) \cdot \ln(2/\delta) \rceil$$