

CSC 284/484 - homework 2 (Monte Carlo)

<http://www.cs.rochester.edu/~stefanko/Teaching/16CS484>

Students that take the course as 484 are required to do **both** 284/484 and 484 parts of the homework. Students that take the course as 284 are only required to do 284/484 part of the homework (of course you are welcome to solve/turn-in the 484 part as well).

1 284/484 homework - solve and turn in

1.1 Theoretical part

Exercise 1.1 (due 2/11/2016) Random variable X has squared coefficient of variation A (that is, $V[X]/E[X]^2 = A$). Random variable Y has squared coefficient of variation B . Let $X_1, \dots, X_n, Y_1, \dots, Y_m$ be independent with X_i 's having the same distribution as X and Y_j 's having the same distribution as Y . Let Z

$$Z = \frac{X_1 + \dots + X_n}{n} \cdot \frac{Y_1 + \dots + Y_m}{m}.$$

What is $E[Z]$? What is the squared coefficient of variation of Z ?

Exercise 1.2 (due 2/11/2016) Let K be a real number. We have two unknown parameters a, b where $0 \leq b - a \leq K$. We are getting samples from the following distribution: Uniform $[a, b]$. We want to output absolute error estimates \hat{a}, \hat{b} of a, b , that is, \hat{a} and \hat{b} should satisfy

$$P(|a - \hat{a}| \leq \varepsilon \wedge |b - \hat{b}| \leq \varepsilon) \geq 1 - \delta.$$

Design a procedure for the estimation that uses as few samples as possible (asymptotically). Prove that your procedure is correct. Clearly state the (asymptotic) number of samples used by your procedure.

1.2 Applied part

Exercise 1.3 (due 2/11/2016) Implement the algorithm that approximately counts the number of satisfying assignments of a DNF formula. Your implementation should read the input from `stdin`. The input is:

- a DNF F given in the following format: the first line contains two numbers n (the number of variables) and m (the number of (conjunctive) clauses); the next m lines contain the clauses of the formula F : a number $i \in \{1, \dots, n\}$ stands for literal x_i , a number $-i \in \{1, \dots, n\}$ stands for literal $\neg x_i$.
- the next line contains precision $\varepsilon > 0$,
- the final line contains error probability $\delta \in (0, 1)$.

The output of your algorithm should be a single number X —your estimate of the number of satisfying assignments of the input formula (the number should be output to `stdout`). Your algorithm should guarantee that $P(X \in [(1 - \varepsilon)C, (1 + \varepsilon)C]) \geq 1 - \delta$ where C is the number of satisfying assignments of F .

Example input:

3 2
1 2 3
-1 -2 -3
0.1
0.01

The DNF formula encoded is $F = (x_1 \wedge x_2 \wedge x_3) \vee (\neg x_1 \wedge \neg x_2 \wedge \neg x_3)$ (the formula has 2 satisfying assignments).

2 484 homework - solve and turn in

2.1 Theoretical part

Exercise 2.1 (due 2/11/2016) We have two unknown parameters a, b , $a < b$. We are getting samples from the following distribution: $\text{Uniform}[a, b]$. We want to output a relative error estimate \hat{c} of $c := b - a$, that is, \hat{c} should satisfy

$$P(|b - a - \hat{c}| \leq \varepsilon |b - a|) \geq 1 - \delta.$$

Design a procedure for the estimation that uses as few samples as possible (asymptotically). Prove that your procedure is correct. Clearly state the (asymptotic) number of samples used by your procedure.

Exercise 2.2 (due 2/11/2016) We have one unknown parameter a . We are getting samples from the following distribution: $\text{Cauchy}[a, 1]$. The probability density function for the distribution is

$$f(x) = \frac{1}{\pi(1 + (x - a)^2)}.$$

We want to output an absolute error estimate \hat{a} of a , that is, \hat{a} should satisfy

$$P(|a - \hat{a}| \leq \varepsilon) \geq 1 - \delta.$$

Design a procedure for the estimation that uses as few samples as possible (asymptotically). Prove that your procedure is correct. Clearly state the (asymptotic) number of samples used by your procedure.

2.2 Applied part

Exercise 2.3 (due 2/11/2016) Implement the determinant estimator for the permanent of zero-one matrices. Your implementation should read the input from `stdin`. The input is:

- an $n \times n$ matrix A given in the following format: the first line contains a numbers n (the dimensions of the matrix); each of the next n lines contain n numbers from $\{0, 1\}$.
- the next line contains precision $\varepsilon > 0$,
- the final line contains error probability $\delta \in (0, 1)$.

The output of your algorithm should be a single number X —your estimate of the permanent of the input matrix (the number should be output to `stdout`). Your algorithm should guarantee that $P(X \in [(1 - \varepsilon)\text{per}(A), (1 + \varepsilon)\text{per}(A)]) \geq 1 - \delta$.

Example input:

```
3
1 1 1
1 1 1
1 1 1
0.1
0.01
```

(Permanent of the input matrix above is 6.)