

Universal methods for derandomization

Lecture 2 Dispersers and Extractors


1

Yesterday

- Five examples of the usefulness of randomness in efficient computation.
- One **universal** task (finding hay in a haystack) which captures at least 2 of them with respect to derandomization.

2

The Simplex Method for Linear Programming




Example 2: Finding Witnesses

- Is $(x+y)(x+y) = (x+x) - (y+y)$?
- Is $(x+y)(x+y)(z-u) + u - (v+u)(x-y) + uz = (y+z)(u-v) + u + (u-v)z + (u-x)$?

Example 3: Monte Carlo Integration

What is the area of A ? Approximately 4.10.



Simulated Annealing

```

SimulatedAnnealing(x, T)
  y := feasible solution to x;
  repeat
    pick a random neighbor z of y;
    y := z with probability
      min(1, exp((k(y)-k(z))/T));
  until (stabilized);
  return the best y found;
    
```

Constructing Hard Truth Tables

- A truth table tabulates $f: \{0,1\}^n \rightarrow \{0,1\}$.

1	1	1	0
0	0	0	0
0	0	1	1
0	1	0	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

A Universal Task: Finding Hay in a Haystack (Black Box version)

Given **black box** $C: \{0,1\}^n \rightarrow \{0,1\}$ with $\mu(C) \geq 1/2$, find x so that $C(x)=1$.

Want: Algorithm polynomial in n .

4

A Universal Task: Finding Hay in a Haystack (Circuit version)

Given **circuit** $C: \{0,1\}^n \rightarrow \{0,1\}$ with $\mu(C) \geq 1/2$, find x so that $C(x)=1$.

Want: Algorithm polynomial in n and size of C .

5

Hypothesis H

There exists deterministic polytime procedure **findHay** taking as input a circuit $C: \{0,1\}^n \rightarrow \{0,1\}$ so that

1. **findHay**(C) is in $\{0,1\}^n$.
2. If $\mu(C) \geq 1/2$ then $C(\mathbf{findHay}(C))=1$.

6

Fact

The constant $\frac{1}{2}$ can be replaced with

- any constant strictly between 0 and 1,
- n^{-k} (quite close to 0), or
- $1-2^{-n^{1-\epsilon}}$ (very close to 1),

without changing truth value of Hypothesis H.

This is not the end of the story – we shall do even better later today!

7

Numerical integration revisited

Density Estimation: Given circuit

$C: \{0,1\}^n \rightarrow \{0,1\}$,

estimate $\mu(C)$ within additive error ϵ .

Desired: Deterministic algorithm running in time polynomial in C and $1/\epsilon$.

8

Finding Hay Derandomizes Monte Carlo Integration

Hypothesis H



An efficient deterministic algorithm for density estimation exists.

9

Lemma

To solve the density estimation problem it is sufficient to make polynomial procedure **Estimate** so that:

For $C: \{0,1\}^n \rightarrow \{0,1\}$,

- $\mu(C) \leq 2^{-n^{1/2}} \Rightarrow$ **Estimate**(C) returns **small**
- $\mu(C) \geq 1 - 2^{-n^{1/2}} \Rightarrow$ **Estimate**(C) returns **big**

10

Reduction

```
ApproximateDensity( $C, \epsilon$ ){
  for( $\alpha=0$  to 1 step  $\epsilon/2$ )
    Let  $C_\alpha =$ 
      [ ( $x_1, x_2, \dots, x_t$ )  $\rightarrow$ 
        if  $\#\{j \mid C(x_j)=1\} / t$  is  $\epsilon/2$ -close to  $\alpha$ 
          then 1 else 0 ]
    If Estimate( $C_\alpha$ ) = big then return  $\alpha$ 
}
```

11

Estimation algorithm, first attempt

```
Estimate( $C: \{0,1\}^n \rightarrow \{0,1\}$ ),
  if  $C(\text{findHay}(C))=1$  return big;
  else return small;
```

12

$\{0,1\}^n$

If C is very small, the union of a small number of random translates of C is still a small set.

13

$\{0,1\}^n$

If C is very big, a small number of random translates of C covers everything with high probability.

14

Lemma (page 15, top)

- Let $C: \{0,1\}^n \rightarrow \{0,1\}$, $C \subseteq \{0,1\}^n$
- Pick y_1, y_2, \dots, y_n at random in $\{0,1\}^n$.
- Let $\bar{C} = \bigcup_i C \oplus y_i$
- $\mu(C) \leq 2^{-n^{1/2}} \Rightarrow \mu(\bar{C}) \leq n2^{-n^{1/2}}$
- $\mu(C) \geq 1 - 2^{-n^{1/2}} \Rightarrow \Pr[\mu(\bar{C}) = 1] > 1/2$

15

Estimation algorithm

Estimate($C: \{0,1\}^n \rightarrow \{0,1\}$),
if $D(\mathbf{findHay}(D))=1$ **return big**;
else return small;

D: Input: y_1, y_2, \dots, y_n in $\{0,1\}^n$.
 Output: 0 if $E(\mathbf{findHay}(E))=1$, 1 otherwise.

E: Input: x in $\{0,1\}^n$.
 Output: 0 if $\exists i: C(x \oplus y_i)=1$, 1 otherwise.

16

Analysis

- The characteristic set of E is the complement of \bar{C}
- If $\mu(C)$ is very small, $\mu(E)$ is very big, no matter what y_1, y_2, \dots are. Hence D always outputs 0 and **Estimate**(C) returns **small**.
- If $\mu(C)$ is very big, $\mu(E)=0$ for more than half the possible values of y_1, y_2, \dots . Hence $\mu(D) > 1/2$ and **Estimate**(C) returns **big**.

17

PrP vs. PrRP

- In the proof, Hypothesis H can be replaced with the hypothesis that we can efficiently **distinguish** between circuits C with $\mu(C)=0$ and circuits C with $\mu(C) \geq 1/2$.
- This is the **PrP = PrRP** assumption discussed in notes.
- It is not known how to get the conclusion assuming only **P=RP** or **P=BPP**.

18

Exercise

If Hypothesis H is true, all randomized approximation algorithms and heuristics can be derandomized:

On input x , the expected quality of solution found by randomized algorithm is q



On input x , the quality of the solution found by deterministic algorithm is $(1+\epsilon)q$

It is not known how to get this assuming only $\text{PrRP}=\text{PrP}$.

19

If Hypothesis H is true, can the construction of hard truth tables be derandomized?

Not so clear.....

But we shall see that the converse is the case! This is a cornerstone in the theory of derandomization: The Impagliazzo-Wigderson 1997 result.

20

Not all applications of the probabilistic method can be derandomized!

- Let \mathbf{R} be the set of **Kolmogorov random strings**: Strings whose **shortest effective description** (i.e. shortest generating program) is almost as long as the string itself.
- A randomly chosen string is in \mathbf{R} with high probability, but no **recursive** procedure can generate an element of \mathbf{R} of length n on input n .

21

Other non-derandomizable settings

- Crypto
- Other Multiparty settings

22

Robustness of Hypothesis H

The constant $\frac{1}{2}$ can be replaced with

- any constant strictly between 0 and 1,
 - n^{-k} (quite close to 0), or
 - $1-2^{-n^{1-\epsilon}}$ (very close to 1),
- without changing truth value of Hypothesis H.

To get even closer to 1, we should try to reduce error probability using few random bits. This question can be considered even in black-box model.

23

Independent Sampling

Let $T: \{0,1\}^n \rightarrow \{0,1\}$. We can find a point x with $T(x)=1$ with

- Random bit usage: $m n$.
- m probes.
- Error probability 2^{-m} .

Can we achieve a better tradeoff?

24

Exhaustive Search

We can find a point x with $T(x)=1$ with

- Random bit usage: 0.
- $2^n/2 + 1$ probes.
- Error probability: 0.

We shall restrict our attention to protocols using $n^{O(1)}$ probes and consider tradeoffs between the other two parameters.

25

Dispersers

- A **Disperser** is an efficient algorithm encoding a particular strategy for finding hay:

$$\mathbf{D}: \{0,1\}^R \times \{1,\dots,m\} \rightarrow \{0,1\}^n$$

- Using R random bits y , the disperser probes $T(\mathbf{D}(y,1)), \dots, T(\mathbf{D}(y,m))$.

- The error probability of the disperser is $\max_{T, \mu(T) \geq 1/2} \Pr_y[\forall i: T(\mathbf{D}(y,i))=0]$

26

“Amplification by Repetition” viewed as disperser

- $R = mn$
- $\mathbf{D}(x_1, x_2, \dots, x_R, i) = x_{n(i-1)+1} x_{n(i-1)+2} \dots x_{n(i-1)+n}$
- Error probability 2^{-m} .

27

Dispersers

	Random bits R	Probes m	Error Prob.
Chor-Goldreich '86	$2n$	$p(n)$	$1/p(n)$
Karp-Pippenger-Sipser '86	n	$p(n)^{O(1)}$	$1/p(n)$
Trevisan '99	$n^{O(1)}$	$n^{O(1)}$	2^{-R+R^ϵ}

28

Chor-Goldreich disperser

- Let $\cdot, +$ be arithmetic operations over $\mathbf{GF}[2^n]$
- $\mathbf{D}((a,b), x) = a \cdot x + b$
- For different x and y and random (a,b) , $a \cdot x + b$ and $a \cdot y + b$ are **independent** random variables.

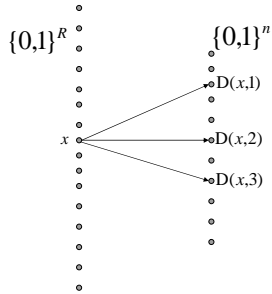
29

Dispersers

	Random bits R	Probes m	Error Prob.
Chor-Goldreich '86	$2n$	$p(n)$	$1/p(n)$
Karp-Pippenger-Sipser '86	n	$p(n)^{O(1)}$	$1/p(n)$
Trevisan '99	$n^{O(1)}$	$n^{O(1)}$	2^{-R+R^ϵ}

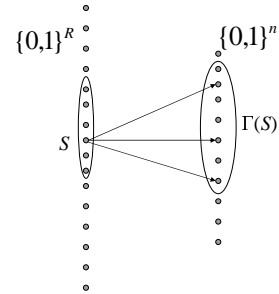
30

Disperser viewed as graph



31

Disperser viewed as graph



32

Dispersion = good expansion

D has error probability $< \epsilon$.
 \Updownarrow
 Every set S of size $\epsilon 2^R$ has $|\Gamma(S)| > \frac{1}{2} 2^n$.

Threshold of disperser: $\epsilon 2^R$

33

Proof

D has error probability $\geq \epsilon$
 \Updownarrow
 $\exists T \subseteq \{0,1\}^n, \mu(T) \geq \frac{1}{2}, \Pr_y[\forall i: D(y,i) \notin T] \geq \epsilon$
 \Updownarrow
 $\exists T \subseteq \{0,1\}^n, \mu(T) \geq \frac{1}{2},$
 $S \subseteq \{0,1\}^R, \mu(S) \geq \epsilon,$
 $T \cap \Gamma(S) = \emptyset$
 \Updownarrow
 $\exists S \subseteq \{0,1\}^R, \mu(S) \geq \epsilon, |\Gamma(S)| \leq \frac{1}{2}$

34

Dispersers

	Random bits R	Probes m	Error Prob.
Chor-Goldreich '86	$2n$	$p(n)$	$1/p(n)$
Karp-Pippenger-Sipser '86	n	$p(n)^{O(1)}$	$1/p(n)$
Trevisan '99	$n^{O(1)}$	$n^{O(1)}$	2^{-R+R^ϵ}

35

Karp-Pippenger-Sipser disperser

- Let G be a constant degree explicit expander graph on $\{0,1\}^n$.
- $D(x, s) = y$ for some s if and only if the distance between x and y in G is at most $c \log n$.

36

Dispersers

	Random bits R	Probes m	Error Prob.
Chor-Goldreich '86	$2n$	$p(n)$	$1/p(n)$
Karp-Pippenger-Sipser '86	n	$p(n)^{O(1)}$	$1/p(n)$
Trevisan '99	$n^{O(1)}$	$n^{O(1)}$	2^{-R+R^ϵ}

37

A physical random bit generator

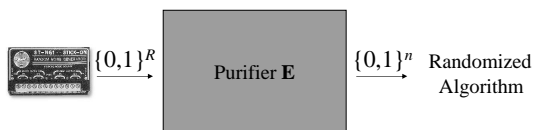


```
0000011000000000000000
01111100000000000000001
0000000000001100000000
000000111100000011010
```

How can we convert bits from a “dirty” random bit source to “pure” random bits?

38

Randomness Purification



Whenever X in $\{0,1\}^R$ “contains much randomness”, $E(X)$ in $\{0,1\}^n$ should be uniformly distributed.

39

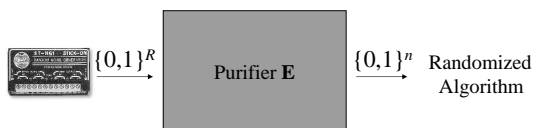
Measure of randomness

Entropy:

$$H(X) = \sum_x -\Pr(X = x) \log \Pr(X = x)$$

40

Randomness Purification



Whenever X in $\{0,1\}^R$ has $H(X) \geq k$, $E(X)$ in $\{0,1\}^n$ should be uniformly distributed.

41

Example: Bit-fixing source

1. An adversary chooses at most $n-k$ “bad” bit positions out of n .
2. The k “good” bit positions are filled in randomly.
3. The “bad” bit positions are filled in by the adversary.
4. The resulting string is given as output without indicating which bits are good and which are bad.

0 0 1 0 1 1 1 0 1 1 1 1

42

Example: Bit-fixing source

1. An adversary chooses at most $n-k$ "bad" bit positions out of n .
2. The k "good" bit positions are filled in randomly.
3. The "bad" bit positions are filled in by the adversary.
4. The resulting string is given as output without indicating which bits are good and which are bad.

0	0	1	0	1	1	1	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---

This is a source of entropy at least k .

43

Randomness Purifiers don't exist!

44

Randomness Purifiers don't exist!

- **Reason 1:** Sources of high entropy may output $00000000\dots000$ with high probability.
 $X = 00000000\dots000$ with probability $\frac{1}{2}$.
 $X =$ uniform on n bits with probability $\frac{1}{2}$.
 $H(X) > n/2$.
- **Solution:** Use a more restrictive notion of randomness-content: **Min-entropy**.

45

Min-entropy

- **Min-entropy** $H^\infty(X) = \min_x -\log \Pr[X=x]$.
- $H^\infty(X)$ is the log of the probability of the **most likely** value of X .
- $H^\infty(X) \leq H(X)$, so it is a more "conservative" estimate of the amount of randomness in a source.
- If f is a deterministic map, $H^\infty(f(X)) \leq H^\infty(X)$.

46

Randomness Purifiers don't exist!

- **Reason 2:** Output of E cannot be uniform on $\{0,1\}^n$ unless probabilities of outcomes of source can be divided into bags of combined measure **exactly** 2^{-n} .
- **Solution:** We shall only demand that the output is **close** to uniform.

47

Variation Distance

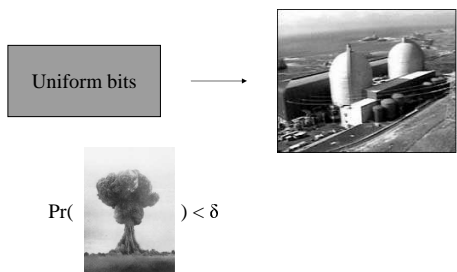
Let X_1 and X_2 be two random variable on the same domain V . The **variation distance** or statistical distance between X_1 and X_2 is

$$\text{dist}(X_1, X_2) = \max_{T \subseteq V} |\Pr[X_1 \in T] - \Pr[X_2 \in T]|$$

The T s are called **statistical tests**.

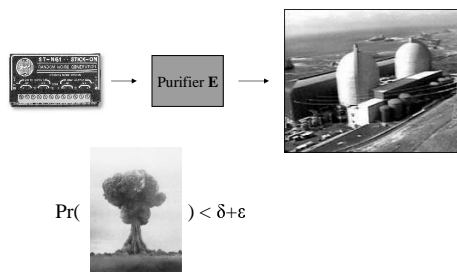
48

Why Variation Distance?



49

Why Variation Distance?



50

Fact

Let $f_1: V \rightarrow [0,1]$, $f_1(x) = \Pr[X_1=x]$.

Let $f_2: V \rightarrow [0,1]$, $f_2(x) = \Pr[X_2=x]$.

$$\text{dist}(X_1, X_2) = \frac{1}{2} \|f_1 - f_2\|_1$$

51

Randomness Purifiers don't exist!

- **Reason 3:** Suppose $E: \{0,1\}^R \rightarrow \{0,1\}^n$ is a purifier. Let y in $\{0,1\}^n$ maximize $|E^{-1}(y)|$. The uniform distribution on $E^{-1}(y)$ has min-entropy $\log |E^{-1}(y)| \geq \log 2^{R-n} = R-n$. But E is constant on this set!
- **Solution:** E needs a **catalyst** of $s \ll n$ pure random bits

52

Extractors

- An **Extractor** extracts randomness from a weak random source:
 $E: \{0,1\}^R \times \{0,1\}^s \rightarrow \{0,1\}^n$
- If X in $\{0,1\}^R$ has min-entropy at least k , U in $\{0,1\}^s$ uniform and independent from X , then $E(X,U)$ is ϵ -close to uniform on $\{0,1\}^n$.
- k is the **min-entropy threshold** of E and ϵ is the **error** of E .

53

Fact

- For restricted classes of sources, the pure random bits are not necessary and we may have **deterministic extractors**.
- Example (Von Neuman): independent, identically distributed random bits.

54

Extractors

- An **Extractor** extracts randomness from a weak random source:

$$\mathbf{E}: \{0,1\}^R \times \{0,1\}^S \rightarrow \{0,1\}^n$$

- If X in $\{0,1\}^R$ has min-entropy at least k , U in $\{0,1\}^S$ uniform and independent from X , then $\mathbf{E}(X,U)$ is ϵ -close to uniform on $\{0,1\}^n$.
- k is the **min-entropy threshold** of \mathbf{E} and ϵ is the **error** of \mathbf{E} .

55

Dispersers

- A **Disperser** is an efficient algorithm encoding a particular strategy for finding hay:

$$\mathbf{D}: \{0,1\}^R \times \{1,\dots,m\} \rightarrow \{0,1\}^n$$

- Using R random bits y , the disperser probes $T(\mathbf{D}(y,1)), \dots, T(\mathbf{D}(y,m))$.

- The error probability of the disperser is

$$\max_{T, \mu(T) \geq 1/2} \Pr_y[\exists i: T(\mathbf{D}(y,i)) = 0]$$

56

Extractors are Dispersers

- An extractor with error $< 1/2$ and min-entropy threshold k is a disperser with error probability $< 2^{-R+k}$.

Proof:

- Suppose not.
- $S \subseteq \{0,1\}^R$ of size $2^{-R+k} 2^R = 2^k$ has $|S| \geq 1/2 2^n$.
- The uniform distribution on S has min-entropy k .
- But $\mathbf{E}(S,U)$ only takes values in S and is hence $1/2$ -far from uniform.

57

- Extractors have many applications where dispersers cannot be used....

58

Monte Carlo Integration

```

volume(A, m) {
  c := 0;
  for(j=1; j<=m; j++)
    if(random(U) ∈ A) c++;
  return c/m;
}

```

59

Randomness-efficient Density Estimation

- Let $m = 10 (1/\epsilon)^2 \log(1/\delta)$. With probability at least $1-\delta$, $\text{volume}(A,m)$ correctly estimates the volume of A within additive error ϵ .
- Can we achieve error probability δ using much less than $\log(|U|) \log(1/\delta)$ random bits? **Not answered by dispersers.....**

60

Fact

- An extractor encodes a strategy for randomness-efficient density estimation (**fairly easy**).
- Conversely, any strategy for randomness efficient density estimation is an extractor (**somewhat harder**).
- As a consequence, the Chor-Goldreich disperser (pairwise independent random variables) is actually a (weak) extractor.

61

Dispersers

	Random bits R	Probes m	Error Prob.
Chor-Goldreich '86	$2n$	$p(n)$	$1/p(n)$
Karp-Pippenger-Sipser '86	n	$p(n)^{O(1)}$	$1/p(n)$
Trevisan '99 Actually Extractor	$n^{O(1)}$	$n^{O(1)}$	2^{-R+R^ϵ}

62