Pairwise independence & derandomization

- a simple randomized algorithm for MaxCut
- pairwise independent sample spaces
- derandomization

Max Cut:

given: $G=(V,E)$

output: partition V into S,T to maximize $\underbrace{\{(u,v) \mid u \in S, v \in T\}}_{\text{size of } S,T \text{ cut}}$ } NP-hard

A randomized algorithm:

Flip n coins $r_1 \cdots r_n$

put vertex i on side r_i to get S,T ← i.e. add i to S if $r_i=0$ & to T o.w.

Analysis:

let $1_{u,v} \equiv \begin{cases} 1 & \text{if } r_u \neq r_v \\ 0 & \text{o.w.} \end{cases}$ (i.e. placed on different sides so (u,v) crosses cut) ⇐ so cut size $= \sum_{(u,v) \in E} 1_{u,v}$

$$E[\text{cut}] = E\Big[\sum_{(u,v)\in E} 1_{u,v}\Big]$$

$$= \sum_{(u,v)\in E} E[1_{u,v}] = \sum_{(u,v)\in E} \Pr[1_{u,v}=1]$$

$$= \sum_{(u,v)\in E} \Pr[(r_u=1 \,\&\, r_v=0) \text{ or } (r_u=0 \,\&\, r_v=1)]$$

$$= \sum_{(u,v)} \Big(\underbrace{\Pr[r_u=1 \,\&\, r_v=0]}_{1/4} + \underbrace{\Pr[r_u=0 \,\&\, r_v=1]}_{1/4}\Big) = \frac{|E|}{2}$$

if $E[cut] = \frac{|E|}{2}$ then $\exists$ cut of size $\geq \frac{|E|}{2}$

why?

- $E[cut]$ is just ave value of cuts coming from random process.
- must be at least one cut which is as big as average value

Pairwise independent random variables: definition

Pick n values $X_1 \cdots X_n$

each $X_i \in T$ (domain) s.t. $|T| = t$ (size of domain)

in some way

def. $X_1 \cdots X_n$ independent if $\forall\, b_1 \cdots b_n \in T^n$

$$\Pr[X_1 \cdots X_n = b_1 \ldots b_n] = \frac{1}{t^n}$$

pairwise independent if $\forall\, i \neq j \quad b_i, b_j \in T^2$

$$\Pr[X_i X_j = b_i b_j] = \frac{1}{t^2}$$

k-wise independent if $\forall$ distinct $i_1 \cdots i_k \quad b_1 \cdots b_k \in T^k$

$$\Pr[X_{i_1} \cdots X_{i_k} = b_1 \cdots b_k] = \frac{1}{t^k}$$

Main point:

(1) Only use pairwise independence in max-cut algorithm

(i.e., algorithm analysis still works if random bits are only pairwise indep).

$\Rightarrow$ if random bits p.i. then $E[\text{cut}] = \frac{|E|}{2}$

$\Rightarrow \exists$ cut chosen by p.i. bits which has size $\geq \frac{|E|}{2}$

(2) Can enumerate over fewer options!!

Derandomization of max-cut

both work pretty well!

Full enumeration :

n fully random bits $\rightarrow \boxed{\text{Algorithm}} \rightarrow$ cut

try all 2^n possible coin tosses
pick best cut

} gets very best cut, not just $\frac{|E|}{2}$

"Partial enumeration":

m pairwise indep random bits $\rightarrow \boxed{\text{Algorithm}} \rightarrow$ cut

don't try all possible coin tosses
just a subset that satisfies pairwise independence

e.g.

pick a row uniformly

r_1	r_2	r_3
0	0	0
0	1	1
1	0	1
1	1	0

for $i \neq j$, $\forall b_1, b_2 \in \{0,1\}^2$

$$\underbrace{\Pr[r_i = b_1 \ \& \ r_j = b_2] = \frac{1}{4}}$$

good enough to give

$$E[\text{cut}] = \frac{|E|}{2} \Rightarrow \exists \text{ cut of size } \frac{|E|}{2} \text{ from this subset of rows}$$

MAIN POINT

for 3 node graphs,
only need to enumerate over 4 rows
instead of 8 rows.

Another picture

enumerate all choices of $r_1 \cdots r_n$

$\boxed{b_1 \cdots b_m} \rightarrow \boxed{\text{"randomness generator"}} \rightarrow \boxed{r_1 r_2 \cdots\cdots r_m}$

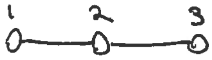
totally independent — enumerate all 2^m choices

$\uparrow$ pick a random row

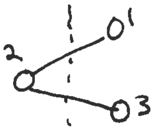
pairwise independent + good enough for our algorithm!

above example: $m = 2$, $n = 3$

CAN WE MAKE $n >> m$?



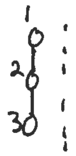
Max Cut:



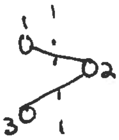
max Value = 2

but we are just claiming to find cut of size $\frac{|E|}{2} = \frac{2}{2} = 1$

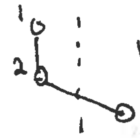
All cuts:



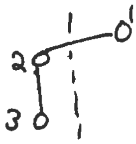
value = 0



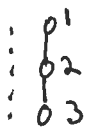
value = 2

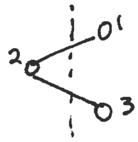


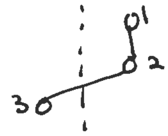
value = 1

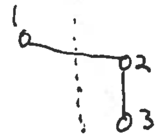


Value = 1







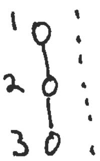


Analysis

$\frac{|E|}{2}$ = Average value: $\frac{2\cdot 0 + 2\cdot 2 + 2\cdot 1 + 2\cdot 1}{8} = 1$ $\Rightarrow \exists$ cut of value

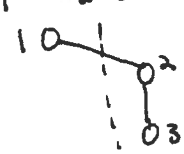
P.I. cuts:

$r_1 = r_2 = r_3 = 0$



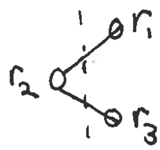
value = 0

$r_1 = 0 \quad r_2 = r_3 = 1$



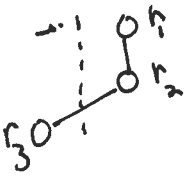
Value = 1

$r_1 = r_3 = 1 \quad r_2 = 0$



Value = 2

$r_1 = r_2 = 1$
$r_3 = 0$



Value = 1

$\frac{|E|}{2}$ = Average value $= \frac{0 + 1 + 2 + 1}{4} = 1$

(same) Analysis $\Rightarrow \exists$ cut of value 1

<u>derandomize Max-Cut</u>, given "randomness generator" taking $(\log n + 1) \Rightarrow n$ bits

- First: construct new randomized MC alg MC': (see picture on next pg)
  - given $\log n$ truly random bits $b_1 \cdots b_{\log n + 1}$
  - use generator to construct n p.i. random bits $r_1 \cdots r_n$
  - Use r_i's in MC alg + evaluate cutsize

- Then: derandomize via enumeration

Deterministic M-C alg:

For all choices of $b_1 \ldots b_{\log n + 1}$

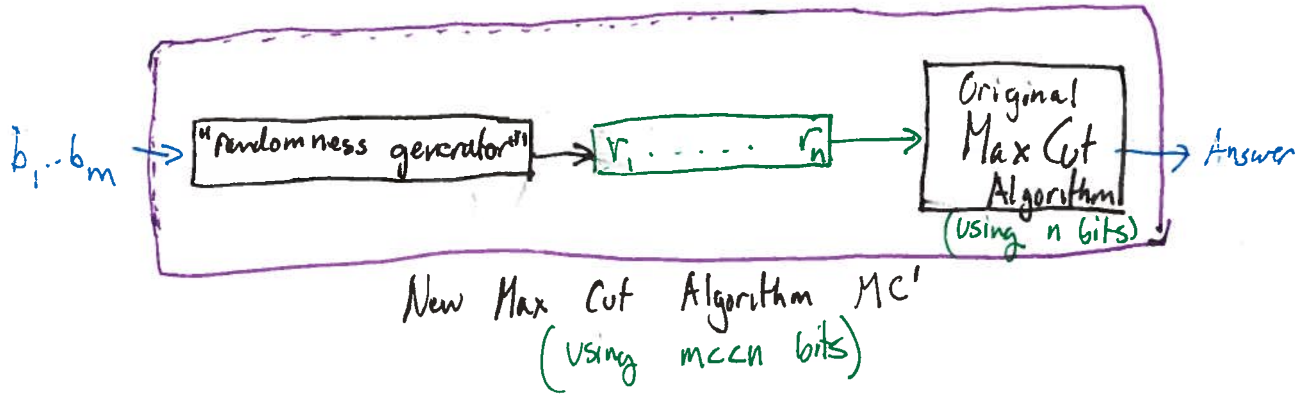
run MC' on $b_1 \ldots b_{\log n + 1}$ + evaluate cutsize

pick best cutsize

Runtime: $\underbrace{\left(2^{c \log n}\right)}_{\#\text{choices of } b_i\text{'s}} \times \left(\text{time for generator} + \text{time to run MC}\right) = \text{poly}(n$

<u>Comments</u>

- no guarantee of getting OPT cut as in basic enumeration method
- generator determines a very small set of random strings, atleast one of which gives a "good" cut



do "full enumeration" derandomization on this in $O(2^m) \times$ [time to generate + time to run MaxCut]

How to generate pairwise independent random variables?

1) Bits

- choose k truly random bits $b_1 .. b_k$

  $\forall S \subseteq [k]$ s.t. $S \neq \emptyset$ set $C_S \equiv \bigoplus_{i \in S} b_i$

- output all C_S

Generates $2^k - 1$ bits from k truly random bits

i.e. $m = \log n$

Generated bits are pairwise independent

proof: exercise

2) Integers in $[0, \dots, q-1]$ (q prime)

trivial method that works for $q = 2^l$ (note that q is not prime)

- repeat "bits" construction independently for each position in $1..l$

uses $O(\log n \cdot \log q) = O(\log^2 n)$ bits of true randomness

Somewhat better construction:

(when $n \approx q$ needs $O(\log q)$ bits of randomness)

- pick $a, b \in \mathbb{Z}_q$
- $r_i \leftarrow a \cdot i + b \mod q \quad \forall\, i \in \{0..q-1\}$
- output $r_1 .. r_q$

Useful to think of as fctn from (input/output description of a)

$$h_{a,b} : [0..q-1] \to \mathbb{Z}_q$$

note: $|\mathcal{H}| = q^2$

Family of fctns $\mathcal{H} = \{h_1, h_2, \ldots\}$ for $h_i : [N] \to [M]$ is

"pairwise independent" if:

when $H \in_u \mathcal{H}$

(1) $\forall\, x \in [N]$, $H(x) \in_u [M]$ — any one location distributed uniformly

(2) $\forall\, x_1 \neq x_2 \in [N]$, $H(x_1)$ + $H(x_2)$ independent — any 2 are indep

notation: "$x \in_u D$" means x chosen uniformly at random from D

equivalently: $\forall\, x_1 \neq x_2 \in [N]$

$\forall\, y_1, y_2 \in [M]$

$$\Pr_{H \in \mathcal{H}}[H(x_1) = y_1 \wedge H(x_2) = y_2] = \frac{1}{M^2}$$

Comments

- no single fctn is p.i. – have to pick a random fctn from a family
- given $\mathcal{H}$ $\forall x \in [N]$ $\mathcal{H}(x)$ should be computable in time poly $(\log N, \log M)$ } don't have to compute "all at once"
- also called "strongly 2-universal hash fctns"

Why is our example p.i.?

$$\mathcal{H} = \{h_{a,b} \mid \mathbb{Z}_q \to \mathbb{Z}_q\}$$ (recall q is prime)

$$h_{a,b} = ax + b \mod q$$

fix any $x \neq w$, c, d

$$\Pr_{a,b}[\overbrace{ax+b}^{h_{a,b}(x)} = c \wedge \overbrace{aw+b}^{h_{a,b}(w)} = d] = \frac{1}{q^2}$$

$$\underbrace{\begin{pmatrix} x & 1 \\ w & 1 \end{pmatrix}}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix}$$

$w \neq x$ so nonsingular } $\Rightarrow$ unique soln

how many truly random bits?

$2 \log q$ yields q p.i. random field elts.

More Comments

- can construct for all finite fields, even when domain + range have different sizes
- original motivation: hashing

  hash fctns chosen from p.i. family instead of random fctns.

  Why is this good?

  how would you store a random fctn on a domain of size $2^{1000000000000000000000\ldots}$