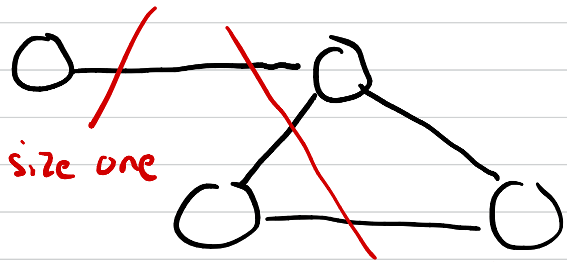
End of Randomized Algorithms

iclickr.com/RGWR

Minimum Cut

Input: Unweighted, undirected graph $G=(V,E)$
Solution: Cut $(C,\bar{C})$ of minimum size



Idea: Use Max-Flow/Min-Cut alg

Computes min s-t cut. Which s, t to use?

Set $s=1$. Try all $t=2,\dots,n$.

Runtime: $n \cdot$ (time for max-flow alg) $\approx n \cdot O(m)$ $n=|V|$, $m=|E|$

100+ pages of math!

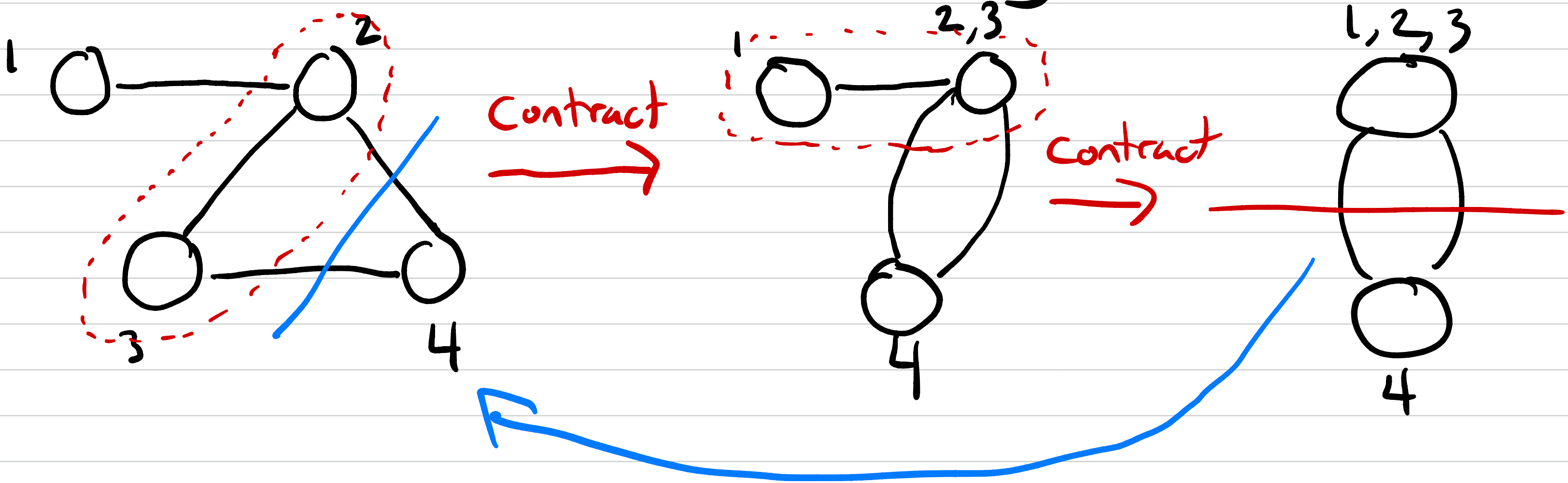
Today: elegant Monte Carlo alg, runs in time $O(n^2)$ for success prob 99%

Karger's algorithm (G)

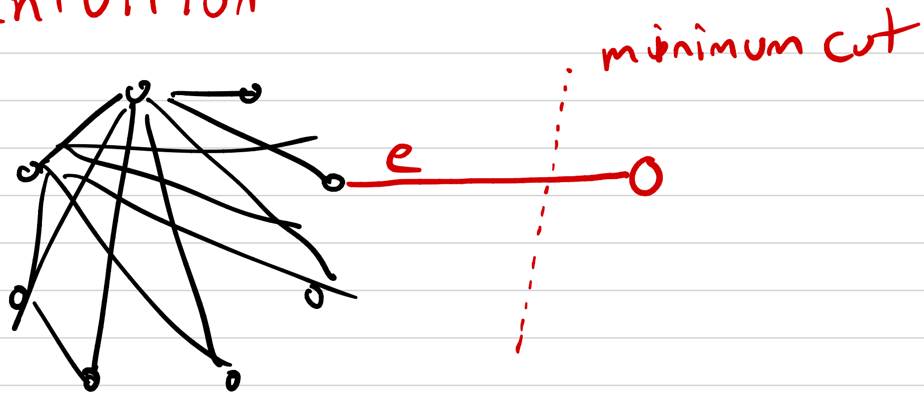
for $i = 1, \ldots, n-2$ $\quad (n = |V|)$

1. pick a uniformly random edge e_i
2. "contract" e_i

return cut specified by the remaining two supervertice



Intuition



Karger's alg finds min cut if it never contracts e

But way more edges on left

Will usually pick there!

More generally: Many edges in graph, few in min cut

Thm: Let $C=(S,\bar{S})$ be a min cut of size k.

$$\Pr[\text{Karger's alg outputs } C] \geq \frac{1}{\binom{n}{2}} = \frac{2}{n(n-1)}$$

Pf: Def: G_i = graph at beginning of i-th iteration. $(G_1 = G)$

H_i = "happy" event the i-th contracted edge e_i doesn't cross cut $C=(S,\bar{S})$

$$\Pr[H_i \mid H_1,\ldots,H_{i-1}] \geq \frac{n-i-1}{n-i+1}$$

$$\begin{aligned}
&\Pr[\text{Alg outputs } (C,\bar{C})] \\
&= \Pr[H_1 \wedge H_2 \wedge \cdots \wedge H_{n-2}] \\
&= \Pr[H_1]\cdot\Pr[H_2 \mid H_1]\cdot\Pr[H_3 \mid H_1,H_2]\cdots\Pr[H_{n-2}\mid H_1,\ldots,H_{n-3}] \\
&\geq \left(\frac{\cancel{n-2}}{n}\right)\cdot\left(\frac{\cancel{n-3}}{n-1}\right)\cdot\left(\frac{\cancel{n-4}}{\cancel{n-2}}\right)\cdots\left(\frac{2}{\cancel{4}}\right)\cdot\left(\frac{1}{\cancel{3}}\right) \\
&= \frac{2}{n(n-1)}
\end{aligned}$$

$$\Pr[H_i \mid H_1, \ldots, H_{i-1}] = 1 - \Pr[\text{Contract edge in } C \mid H_1, \ldots, H_{i-1}]$$

$$= 1 - \frac{\#\text{ edges across } C}{\#\text{ edges in } G_i}$$

$$\geq 1 - \frac{k}{\frac{1}{2}k(n-i+1)} = 1 - \frac{2}{n-i+1} = \boxed{\frac{n-i-1}{n-i+1}}$$

close to 1 if $i=1$, $= \frac{1}{3}$ if $i = n-2$

Fact 1: $\text{Min-Cut}(G_i) \geq k$

(any cut in G_i corresponds to cut in G)

Fact 2: # vertices in $G_i = n-(i-1) = n-i+1$

Fact 3: degree of each vertex in $G_i \geq k$

Fact 4: # of edges in $G_i = \frac{1}{2}\sum_{v \in G_i} \deg(v)$

$$\geq \frac{1}{2}\sum_{v \in G_i} k$$

$$= \frac{1}{2} \cdot k \cdot |G_i| \geq \frac{1}{2} \cdot k \cdot (n-i+1)$$

$\Pr[\text{succeeds}] \geq \frac{1}{\binom{n}{2}} \approx \frac{2}{n^2}$

Succeed w/ constant prob: repeat n^2 times

(or slightly more)

Fact: # of min cut $\leq \binom{n}{2}$

$\Pr[\text{output min cut}] \geq \sum_{\text{min cut}} \frac{1}{\binom{n}{2}} = \frac{(\text{\# of min cuts})}{\binom{n}{2}}$.

Online

Algorithms

San Antonio Express-News

Football experts

Game	 Columnist **Mike Finger**	 Sports reporter **Tim Griffin**	 Sports columnist **Buck Harvey**	Me	Outcome
Cowboys v. 49ers	Cowboys	49ers mistake	49ers mistake	49ers mistake	Cowboys
Loss	0	1	1	1	(Penalty for bad decision)
Cowboys v. Steelers	Steelers mistake	Cowboys	Steelers mistake	Steelers mistake	Cowboys
Loss	1	0	1	1	
Cowboys v. Patriots	?	?	?	?	

Goal: Predict as well as ~~possible~~ the best expert.

Experts problem

There are n experts $E_1, \ldots, E_n$ and T days.
On each day $t = 1 \ldots T$:
1. Each expert makes a prediction (e.g. Yes/No)
2. You make a prediction (e.g. YES/NO)
3. The real outcome is revealed

Goal: At time T, (# of mistakes you make) $\leq$ (# of mistakes of best experts) + (small)

Applications: stock market, sports predictions, etc.

Features of model

1. ~~Not~~ assuming past predictions predict future performance. (all algs useless!?!?)
2. Outcomes can be chosen adversarially, based on past predictions, even current prediction!

3. Input provided piece-by-piece

A simple case

Promise: one of the n experts $E_1, \ldots, E_n$ never makes a mistake

Goal: make fewest mistakes possible

Idea: never follow expert who's already made a mistake

Alg 1: Follow 1^{st} expert until they make a mistake,
Then 2^{nd} " " " " " " , and so on.

Can make $n-1$ mistakes.

Alg 2 ("Halving algorithm"): For day $t = 1, \ldots, T$:
1. Set $\text{Correct}_t \subseteq [n]$ to be the set of experts who have not yet erred
2. Predict Yes if $\geq 1/2$ of Correct_t experts do so
Predict No otherwise

Thm: Halving makes $\leq \log_2(n)$ mistakes.

Pf: When Halving makes mistake, $|\text{Correct}_{t+1}| \leq \frac{1}{2} \cdot |\text{Correct}_t|$. Can only happen $\log_2(n)$ times. $\square$

A harder case

Promise: one of the n experts makes $\leq M$ mistakes

Alg: For day $t = 1 \ldots T$

1. Let m^t_i = # mistakes i-th expert has made so far
2. Set $\text{Plausible}_t \subseteq [n]$ to be the experts who have made $\leq M$ mistakes so far
3. Predict majority vote of Plausible_t experts

Thm: This Alg can make $\geq (M+1) \cdot \log(n)$ mistakes

Pf:

Experts / Days	$\overbrace{1 \quad 2 \quad \ldots}^{\text{half}}$	$\overbrace{\qquad\qquad}^{\text{half}}$	n	Alg	Real outcome
1	Y Y ... Y			Y	N
⋮				Y	N
M+1	Y Y ... Y			Y	N
M+2		Y Y ... Y		Y	N
⋮				Y	N
				Y	N
2M+2		Y Y ... Y	⋮	Y	N

Can continue for $(M+1)\log_2(N)$ rounds $\square$

Goal: (# of mistakes) $\leq$ (best expert) + (small)

Weighted majority alg (given parameter $0<\varepsilon<1$)

Initialize weights $w_1^{(1)}, \dots, w_n^{(1)} = 1$

For $t = 1, \dots, T$:

1. Output weighted majority of experts according to $w^{(t)}$ weights.
2. Set $w_i^{(t+1)} = (1-\varepsilon) \cdot w_i^{(t)}$ if i-th expert erred.

   Set $w_i^{(t+1)} = w_i^{(t)}$ otherwise.

Example w/ $\varepsilon = 1/2$:

		Experts 1	2	3	4	Alg	Real
Day 1:	Weights	1	1	1	1		
	Predictions	Y	Y	Y	N	Y	N
Day 2:	W	1/2	1/2	1/2	1		
	P	Y	N	N	Y	Y	N
Day 3:	W	1/4	1/2	1/2	1/2		

Suppose we run WM alg w/ $\varepsilon = 1/2$.

Fact 1: Let $W^{(t)}$ = total weight of experts at time t.
Suppose WM made a mistake at time t.
Then $W^{(t+1)} \leq \underline{3/4} \cdot W^{(t)}$. (Halve $\geq 1/2$ the weight)

Fact 2: Suppose expert i makes m_i mistakes.
Then $w_i^{(T+1)} \leq \underline{(1/2)^{m_i}}$.

Thm: Let M = # mistakes WM alg makes w/ $\varepsilon = 1/2$.
Then $M \leq 2.4 \cdot (OPT + \log_2(n))$.

Pf: Let i^* = best expert.
Then $w_i^{(T+1)} = (1/2)^{OPT}$. So $W^{(t+1)} \geq (1/2)^{OPT}$.

Alg makes M mistakes $\Rightarrow W^{(t+1)} \leq \left(\frac{3}{4}\right)^M \cdot W^{(1)} = \left(\frac{3}{4}\right)^M \cdot n$

$$\therefore \left(\frac{1}{2}\right)^{OPT} \leq \left(\frac{3}{4}\right)^M \cdot n \Rightarrow \underbrace{\left(\frac{4}{3}\right)^M}_{2^{\log_2(4/3)\cdot M}} \leq 2^{OPT} \cdot n \Rightarrow M \leq 2.4 \cdot (OPT + \log_2(n)). \ \square$$

Can't beat 2

		Experts		
E_1	E_2	Alg	Real = opposite of alg
1	0	0	1
1	0	1	0
1	0	1	0
1	0	0	1
	⋮		
1	0		

\# of Alg mistakes = T
\# of mistakes of best expert $\leq T/2$

$\therefore$ (# of Alg mistakes) $\geq 2 \cdot OPT$

The fix: Let Alg be randomized.
Alg's prediction is probability distribution (p_0, p_1).
Loss at day t is p_0 if real outcome is 1,
p_1 " " " " 0.

Randomized weighted majority (w/ parameter ε)

Initialize weights $w_1^{(1)}, \dots, w_n^{(1)} = 1$

For $t = 1, \dots, T$:

1. Set $W^{(t)} = w_1^{(t)} + \dots + w_n^{(t)}$
2. Guess $b \in \{0,1\}$ w/prob $\frac{1}{W^{(t)}} \cdot \sum_{i=1}^{n} w_i^{(n)} \cdot \mathbb{1}[E_i \text{ guesses } b \text{ today}]$
3. Set $w_i^{(t+1)} = (1-\varepsilon) \cdot w_i^{(t)}$ if i-th expert erred.

   Set $w_i^{(t+1)} = w_i^{(t)}$ otherwise.

Thm: Set $M = \#$ mistakes. Then $\mathbb{E}[M] \leq \overbrace{(1+\varepsilon) \cdot OPT + \frac{1}{\varepsilon} \log_2(n)}^{OPT + \varepsilon \cdot \overset{\leq T}{\cancel{OPT}} + \frac{1}{\varepsilon} \cdot \log_2(n)}$

Set $\varepsilon = \sqrt{\frac{\log_2(n)}{T}}$. Then $\mathbb{E}[M] \leq OPT + 2\sqrt{T \log_2(n)}$

So $\mathbb{E}[M]/T \leq OPT/T + \boxed{2\sqrt{\log_2(n)/T}} \xrightarrow{T \to \infty} 0$

"no regret algorithm"

General setting

There are n actions you can take each step (corresponding to n experts)

On day t, you play action $i_t \in [n]$,

action i has cost $c_i^{(t)} \in [0,1]$,

Goal: (Alg's cost) $\leq$ (OPT action) + small

$$\sum_{t=1}^{T} c_{i_t}^{(t)} \leq \min_{i^*} \left\{ \sum_{t=1}^{T} c_{i^*}^{(t)} \right\} + \boxed{\text{small}} \longrightarrow \text{Your "regret"}$$

Multiplicative Weights Algorithm

Initialize weights $w_1^{(1)}, \dots, w_n^{(1)} = 1$

For $t = 1, \dots, T$

1. Set $W^{(t)} = w_1^{(t)} + \cdots + w_n^{(t)}$
2. Play action i w/prob $w_i^{(t)} / W^{(t)}$
3. Observe costs $c_1^{(t)}, \dots, c_n^{(t)}$
4. Set $w_i^{(t+1)} = w_i^{(t)} \cdot (1 - \varepsilon \cdot c_i^{(t)})$

Thm: Set $\varepsilon = \sqrt{\log_2(n)/T}$.

Then $E[\text{Regret}] \leq O(\sqrt{T \log_2(n)})$

$\therefore$ $E[\text{Regret}]/T \leq O(\sqrt{\log_2(n)/T})$ "no regret"

Strong duality of zero-sum games

Let $M = \begin{bmatrix} M_{11} & & M_{1n} \\ \vdots & \ddots & \vdots \\ M_{n1} & & M_{nn} \end{bmatrix}$ be an $n \times n$ payoff matrix, with $-1 \le M_{ij} \le 1$.

Given $p = \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix}$ and $q = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix}$, $\text{Score}(p,q) = \sum_{i,j} p_i M_{ij} q_j = p^T M q$.

Thm: $\max_p \min_q p^T M q = \min_q \max_p p^T M q$.

Pf: "Self play": Row and Column players T games, each time improving their strategies p and q.

$p^{(t)}$ = Row player's strategy in t-th game, generated by multiplicative weights alg. (where n experts = n rows of M)

$q^{(t)}$ = Defined similarly

Row | Col

Game t:

Row:

1. MW alg outputs $p^{(t)}$
2. Give MW alg loss vector $-Aq^{(t)}$
3. Total loss $= -(p^{(t)})^T A q^{(t)}$
   $= -\text{Score}(p^{(t)}, q^{(t)})$

Col:

1. MW alg outputs $q^{(t)}$
2. Give MW alg loss vector $A^T p^{(t)}$
3. Total loss $= (p^{(t)})^T A q^{(t)}$

MW alg guarantee:

$$\text{(Row)} \quad \frac{1}{T}\sum_{t=1}^{n}(p^{(t)})^T(-A)q^{(t)} \le \min_i \frac{1}{T}\sum_{t=1}^{T}\left((-A)q^{(t)}\right)_i + 4\sqrt{\frac{\ln(n)}{T}}$$

$$\text{(Col)} + \quad \frac{1}{T}\sum_{t=1}^{n}(p^{(t)})^T A q^{(t)} \le \min_j \frac{1}{T}\sum_{t=1}^{T}\left(A^T p^{(t)}\right)_j + 4\sqrt{\frac{\ln(n)}{T}}$$

$$0 \le \min_i \frac{1}{T}\sum_{t=1}^{T}\left((-A)q^{(t)}\right)_i + \min_j \frac{1}{T}\sum_{t=1}^{T}\left(A^T p^{(t)}\right)_j + 8\sqrt{\frac{\ln(n)}{T}}$$

$$0 \le \min_i \frac{1}{T}\sum_{t=1}^{T} \left((-A)q^{(t)}\right)_i + \min_j \frac{1}{T}\sum_{t=1}^{T} \left(A^T p^{(t)}\right)_j + 8\sqrt{\frac{\ln(n)}{T}}$$

Def: $p_{avg} = \frac{1}{T}\sum_{t=1}^{T} p^{(t)}$ and $q_{avg} = \frac{1}{T}\sum_{t=1}^{T} q^{(t)}$

$$0 \le \underbrace{\min_i \left((-A)q_{avg}\right)_i}_{\left(\begin{array}{l} -\max_i (Aq_{avg})_i \\ = -\max_p p^T A q_{avg} \end{array}\right)} + \underbrace{\min_j \left(A^T p_{avg}\right)_j}_{\left(\min_q p_{avg}^T A q\right)} + 8\sqrt{\frac{\ln(n)}{T}}$$

$$\Rightarrow \min_q \max_p p^T A q \le \max_p (p)^T A q_{avg} \le \min_q p_{avg}^T A q + 8\sqrt{\frac{\ln(n)}{T}}$$

$$\le \max_p \min_q p^T A q + 8\sqrt{\frac{\ln(n)}{T}}$$

$(T \to \infty)$

$$\Rightarrow \min_q \max_p p^T A q \le \max_p \min_q p^T A q. \qquad \square$$