Minimax eljárás

Az előzőekben ismertetett algoritmus várható futási ideje legalább $n^{0,793}$ bármely n levéllel rendelkező egyenletes bináris ÉS-VAGY fa esetén. De van e ennél gyorsabb véletlenített algoritmus? A futási időre vonatkozó alsó korlát igazolására szolgál a következő technika: a minimax eljárás. (Ez az egyetlen ismert általános technika ilyen állítások igazolására). Először a játékelmélettel ismerkedünk.

Játékelméleti alapfogalmak

Roberta és Charles kő-papír- ollót játszik. A vesztes 1$-t fizet a győztesnek. A végeredmény azonos jel esetén számolódik. Ez a játék egy mátrixszal bemutatható.

		Charles		
		Olló	Papír	Kő
Roberta	Olló	0	1	-1
	Papír	-1	0	1
	Kő	1	-1	0

C által R-nek fizetett értékek vannak a mátrixban

Ez egy példa a 2 személyes zéróösszegű (a nyeremények összege 0) játékokra, és ezt a mátrixot kifizetési mátrixnak hívjuk. Minden 2 személyes zéróösszegű játék reprezentálható egy valós értékekkel rendelkező n x m -es mátrixszal: **C**. (megj: **C**: mátrix; **m**: vektor) A sorok az R játékos lehetséges stratégiáit tartalmazzák, az oszlopok pedig C-ét. A $\mathbf{C_{ij}}$ érték a C által választott j, és az R által választott i stratégia által létrejött kifizetés. Persze R célja a sorok alapján a bevétel maximalizálás, ill. a kiadás minimalizálása. Persze, egyik fél sem rendelkezik információval a másik fél stratégiáját illetően. (zéróinformációs játék)
R által választott i stratégia szerint számára garantálva van egy kifizetés $\min_j \mathbf{C_{ij}}$ alapján, függetlenül C stratégiájától. R számára az optimális stratégia maximalizálja $\min_j \mathbf{C_{ij}}$ -t. Legyen $V_R = \max_i \min_j \mathbf{C_{ij}}$ jelölje R számára fizetett érték alsó határát, amikor optimális stratégiát használ. C számára az a j stratégia az optimális, ami a lehető legnagyobb lehetséges (C → R) kifizetés felső határát adja. Hasonlóan C optimális stratégiája biztosítja, azt, hogy C kifizetése R-nek legfeljebb $V_C = \min_j \max_i \mathbf{C_{ij}}$.
Minden kifizetési mátrixra igaz a következő egyenlőtlenség:

$$\max_i \min_j \mathbf{C_{ij}} \leq \min_j \max_i \mathbf{C_{ij}}$$

i: sor, j: oszlop

A fenti játékban ezek az értékek: $V_R = -1$, $V_C = 1$. Ha ez a két érték megegyezik, akkor a játéknak van megoldása (nyeregpont) ami $V = V_R = V_C$. Az optimális stratégiákat R és C számára jelezze ekkor $V = \mathbf{C_{\rho\gamma}}$. Általánosságban elmondható, hogy egy játékosnak több optimális stratégiája is lehet, mint egy.
Ennek a játéknak egy módosított változata a következő:

		Charles		
		Olló	Papír	Kő
Roberta	Olló	0	1	2
	Papír	-1	0	1
	Kő	-2	-1	0

A módosított játékban V=0, ρ=1, γ=1 lesz.
Mi van ha játéknak nincs megoldása? Ekkor egyik játékosnak sincs egyértelmű optimális stratégiája. Valójában, az ellenfél stratégiáját ismerve, azt felhasználhatjuk a kifizetés javításához, ellentétben azzal az esettel, amikor a játéknak nyeregpontja van. Érdekes lenne ezt a problémát körbejárni a stratégiák véletlen kiválasztásával.

Nézzük a mixelt stratégiákat. Ez egy valószínűségi eloszlás lehetséges stratégiák halmazán. A sorjátékos választ egy vektort $\mathbf{p} = (p_1,\ldots,p_n)$, ami $\mathbf{C}$ sorainak egy eloszlása: p_i annak a valószínűsége, hogy R által az i-edik sor a választott stratégia, hasonlóan az oszlop játékos vektora: $\mathbf{q} = (q_1,\ldots q_n)$. A kifizetés most egy véletlen változó, és a várható értéke:

$$\mathbf{E}[\text{kifizetés}] = \mathbf{p}^T\mathbf{C}\mathbf{q} = \sum_{i=1}^{n}\sum_{j=1}^{m} p_i\mathbf{C}_{ij}q_j$$

Példaként vegyük a következő játékot:

		B	
A		0	1
		1	0

Ekkor ½ -ed valószínűséggel lehet választani mindkét stratégiát. Itt mind B mind A számára a várható nyereség soronként és oszlopokként: ½ * 0 + ½ * 1 = ½ , tehát B-nek 2 oszlopa miatt ½ + ½ = 1 a teljes várható nyereség, akárcsak A-nak.

V_R -t arra használjuk, hogy kijelöljük a lehetséges legjobb értéket R számára, V_C -t arra használjuk, hogy kijelöljük a lehetséges legjobb értéket C számára:

$$V_R = \max_p \min_q \mathbf{p}^T\mathbf{C}\mathbf{q}$$
$$V_C = \min_q \max_p \mathbf{p}^T\mathbf{C}\mathbf{q}$$

Ezen értékekre teljesül (a lineáris programozás erős dualitási tételével ekvivalens) a következő állítás:

Neumann Minimax Tétel: Bármely 2 személyes zéróösszegű C mátrix által adott játékra:

$$\max_p \min_q \mathbf{p}^T\mathbf{C}\mathbf{q} = \min_q \max_p \mathbf{p}^T\mathbf{C}\mathbf{q}$$

Azaz, a legnagyobb R által, mixelt stratégia választásával, garantált kifizetés megegyezik a legkisebb C által, mixelt stratégia választásával, garantált kifizetéssel. Ezt az értéket jelöljük V-

vel. A két stratégia $(\hat{p},\hat{q})$ egyaránt garantálja a baloldal maximumát, és a jobb oldal minimumát, az érték a nyereg pont, és a két vektor az optimális mixelt stratégiák.
Ha **p** fixált, akkor $\mathbf{p}^T\mathbf{Cq}$ egy lineáris függvénye **q**-nak, és minimalizált az által, hogy az a q_j 1 -re van állítva, amihez a legkisebb együttható tartozik ebben a lineáris függvényben. Tehát ha C ismeri R **p** eloszlását, akkor az ő optimális stratégiája tiszta lehet. Ez fordítva is igaz. Ez a minimax tétel egyszerűsített változatához vezet. Legyen $\mathbf{e}_k$ egy egységvektor, 1-essel a k-adik pozícióban, a többi érték legyen 0. Így a következő tételt kapjuk:

Loomis Tétel: Bármely 2 személyes zéróösszegű, C mátrix által adott játékra:

$$\max_p \min_j \mathbf{p}^T\mathbf{Ce}_j = \min_q \max_i e_i^T\mathbf{Cq}$$

Yao technika

Legyenek a játékot leíró mátrix sorai a rögzített méretű bemenetek, és az oszlopai a lehetséges determinisztikus algoritmusok, az oszlopjátékos C az algoritmus készítője, míg a sorjátékos R generálja a bemenetet. Minden oszlophoz determinisztikus algoritmus tartozik, mely jó megoldást ad. C kifizetése R-nek, nem más mint az algoritmus költsége (ennek mértékegysége bármi lehet, például: költség, távolság, futási idő, stb). C fizetésminimalizáló algoritmust szeretne, míg az ellenfél maximalizálni szeretné azt.

Ekkor C tiszta stratégiája egy determinisztikus algoritmus, míg R számára ez egy lehetséges bemenet. C optimális tiszta stratégiája esetén V_C a legrosszabb futásidejű determinisztikus algoritmus, amit a probléma determinisztikus komplexitásának hívunk. Mixelt stratégiák pedig a véletlenített algoritmusoknak felelnek meg, ebben az esetben C számára az optimális stratégia az optimális LasVegas algoritmus. Tehát a fenti játékelméleti tételek alapján:

Következmény
Legyen Π egy probléma véges számú, rögzített méretű *I* inputtal, és véges számú determinisztikus algoritmussal: *A* . Legyen $I \in I$, $A \in A$, legyen C(I,A) az algoritmus futásideje. Egy *I* feletti lehetséges **p** eloszlással, és *A* feletti lehetséges **q** eloszlással legyen I_p egy véletlen választott input **p**-nek megfelelően, A_q pedig véletlenített algoritmus **q**-nak megfelelően. Ekkor

$$\max_p \min_q \mathbf{E}[C(I_p, A_q)] = \min_q \max_p \mathbf{E}[C(I_p, A_q)]$$

$$\max_p \min_{A \in A} \mathbf{E}[C(I_p, A)] = \min_q \max_{I \in I} \mathbf{E}[C(I, A_q)]$$

A második egyenlőség alapján adódik, hogy minden *I* fölötti **p** és *A* fölötti **q** eloszlásra:

$$\min_{A \in A} \mathbf{E}[C(I_p, A)] \le \max_{I \in I} \mathbf{E}[C(I, A_q)]$$

Tehát azt kaptuk, hogy az optimális determinisztikus algoritmus várható futásideje, egy tetszőlegesen választott **p** input eloszlásra, egy alsó korlátot ad az optimális Las Vegas véletlenített algoritmus várható futásidejére.

Alsó korlát a JátékFa Kiértékeléshez

Yao technikáját fogjuk használni. 2 gyerekes, k szintes T nevű AND-OR fákról lesz szó.
Elsőként vegyük észre, hogy ez a fa ekvivalens egy 2k szintes kiegyensúlyozott bináris fával, ahol az összes belő csúcs NOR logikai kiértékelő pont.
Az alsó korlát igazolásához adnunk kell a levélértékeknek egy eloszlását, és igazolnunk kell egy alsó korlátot az optimális determinisztikus algoritmusnak az input eloszláson várható futásidejére.
Legyen $p= \frac{\left(3 - \sqrt{5}\right)}{2}$. A fa minden levelét p valószínűséggel állítsuk 1 re. Ha minden NOR az inputnál függetlenül 1 p valószínűséggel, akkor annak a valószínűsége, hogy ennek a kimenete is 1, az olyan valószínűségű, mint amikor mindkettő bemenet 0 , ami:

$$\left(\frac{\left(\sqrt{5} - 1\right)}{2}\right)^2 = \frac{\left(3 - \sqrt{5}\right)}{2} = p$$

Tehát a NOR fa minden csomópontja p valószínűséggel 1, és egy csomópont értéke független a többi azonos szinten levő csomópont értékétől. Továbbá vegyük észre, hogy az adott szinten levő pontok függetlensége miatt az az algoritmus optimális, amely mélységi keresés alapján értékeli ki a fát (természetesen, ha egy szinten egyetlen gyerek alapján megkaphatja az értéket, a másikat már nem nézi meg). Ezt az algoritmust mélységi vágás algoritmusnak hívjuk.

Legyen a mélységi vágás algoritmussal kiértékelve a fát, W(h) azon levelek várható száma, amelyeket meg kell néznünk egy h távolságban levő csomópont kiértékelésében. Ekkor: W(h) = W(h - 1)+ (1 - p) x W(h-1). Ha az első tag kiértékelése 0 eredményt ad, akkor a második tagot is ki kell értékelni, ennek 1-p a valószínűsége. Legyen $h = \log_2 n$, ekkor megoldva a rekurziót, kapjuk: $W(h) \geq n^{0,694}$.

Tétel: Bármely véletlenített algoritmusnak (mely mindig helyesen értékeli ki a $T_{2,k}$ bemeneteket – 2 gyerekes, k szintű fa) várható futási ideje legalább $n^{0,694}$, ahol $n = 2^{2k}$ a levelek száma.

Ez az alsó korlát kisebb mint a felső korlát ($n^{0,793}$), ami az előző előadáson vett tételből jön. Elképzelhető hogy ez az alsókorlátos technika gyengébb? Valószínűleg nem a legjobb lehetséges eloszlást választottuk a fa leveleinek. Ha egy NOR csomópont mindkét gyereke/bemenete 1, akkor nincs értelme mindkét részfa leveleit megvizsgálni. Szóval a legjobb alsó korlát bebizonyításához, olyan eloszlás kell amelyben a levelek értékeit véletlenül, de nem egymástól függetlenül választjuk. Ez az erősebb analízis mutatja, hogy az előző órán vett foglalt algoritmus optimális. ($T_{2,k}$ fára a véletlenített algoritmus lépésszáma legalább 3^k.)

Algebrai technikák

Elsőként az ujjlenyomat technikát ismertetjük, ami a következő ötleten alapul. A feladatunk, hogy egy adott U univezumba tartozó x és y elemek megegyeznek-e. Ennek determinisztikus komplexitása log|U|. A véletlenített módszerben veszünk egy véletlen leképezését U-nak egy kisebb elemszámú halmazba, legyen ez V. Ha x és y U-ban megegyezett, akkor V-ben is megegyezik (fordítva ez nem biztos hogy igaz), ennek tesztelésének ideje log|V|. Azt mondjuk V-ben x és y képe az ujjlenyomatuk.

A továbbiakban egy **F** számtest felett vesszük a számokat. Ha **F** véges test, akkor legyen egy p prímnek a maradékosztályaiból álló test.
Többször fogjuk használni a halasztott döntés elvét, amelyet a következő példán mutatunk be.

Adott 1 pakli francia kártya, 52 lap: ebből 13db 4 kártyát tartalmazó kupacot hozunk létre. A következő játékot játsszuk.
1. 13 pakliból kiválasztunk 1et és levesszük a legfelső kártyát
2. amilyen szám van rajta azzal a paklival folytatjuk.
3. Vége: ha üres kupachoz érkezünk. Nyertünk: ha minden kártyát felvettünk

Az a kérdés mi a nyerés valószínűsége, ha egyenletes keverés alapján lettek lerakva a kártyák.
Halasztott döntés elve: az, hogy amikor a következő kártyát felvesszük, akkor az a kártya a még játékban lévő kártyákból egyenletes eloszlás alapján kerül ki. (Úgy vesszük a játékot, mintha a keverést elhalasztanánk eddig az időpontig.)
Következésképpen a játékban egy egyenletes eloszlás alapján kapott véletlen permutáció sorrendjében vesszük a kártyákat. Vegyük észre, hogy pontosan akkor nyerünk, ha király a permutációban az utolsó lap (pontosan a negyedik királynál áll meg a játék). Ennek a valószínűsége, így a játék megnyerésének a valószínűsége is 1/13.

Ujjlenyomat és a Frievald technika

Vegyük elsőként a mátrix szorzás problémáját, amely esetén a legjobb ismert algoritmus időigénye $O(n^{2,376})$, de az nagyon komplikált. Az ezt megvalósító programot ellenőrizhetnünk kell, hogy helyesen számolt e. (Programverifikálás eljárásaihoz hasonlóan).
Legyenek **A**,**B**,**C** nxn-es mátrixok egy **F** számtest fölött. Az AB = C egyenlőséget szeretnénk ellenőrizni. A Freivald technika $O(n^2)$ idő alatt korlátos hibahatárral megoldást ad erre a problémára. Ez a véletlenített algoritmus először választ egy vektort $\mathbf{r} \in \{0,1\}^n$, **r** minden eleme azonosan és függetlenül választott véletlenül 0 vagy 1-nek, amelyek **F** additív és multiplikatív egységelemei. Ki kell számolnunk **x= Br** , **y=Ax = ABr**, és **z= Cr** -t, ezt $O(n^2)$ időben megtehetjük. Nyilvánvalóan, ha **AB=C** akkor **y=z**.
Azt állítjuk, hogy ha **AB≠C** , akkor **y≠z** legalább ½ valószínűséggel. Az algoritmus akkor hibázik ha **y=z** mégis teljesül.

Tétel: Legyen **A**,**B** és **C** n x n -es mátrixok **F** fölött, és legyen **AB≠C**. Válasszunk egy **r**-t egyenletes eloszlás alapján a $\{0,1\}^n$ halmazból. Ekkor **Pr**[**ABr=Cr**] ≤ ½
Bizonyítás: Legyen **D=AB-C**, ekkor **D** nem azonosan 0 mátrix. Feltehetjük, hogy az első sora nem 0, és a nem 0 értékek, megelőzik a 0 értékeket. Annak a valószínűségét szeretnénk meghatározni, hogy **Dr=0**.
Legyen **d** a **D** első sorvektora, melynek első k értéke nem 0, k>0. Vizsgáljuk annak a valószínűségét, hogy **d** és **r** belső szorzata nem 0, ez alsó korlátot ad arra, valószínűségre nézve, hogy **Dr=0** azaz **y=z**.
A belső szorzat $\mathbf{d}^T\mathbf{r}$=0, csak ha :

$$r_1 = \frac{-\sum_{i=2}^{k} d_i r_i}{d_1}.$$

Felhasználhatjuk a halasztott döntés elvét. Minden **r**-ben szereplő komponenst $\mathbf{r}_1$ előtt választunk. Ezzel a fenti formula jobboldala fixálva lesz, ami egy $v \in \mathbf{F}$. Ekkor r_1 egy 2 elemű halmaz fölött egyenletesen generált elem, így annak valószínűsége, hogy ez v-vel megegyezik legfeljebb ½. Ezzel a tételt igazoltuk.

Polinomok azonosságának ellenőrzése

Freivald technikája erre a feladatra is alkalmas. Két polinom P(x) és Q(x) azonos ha a megfelelő együtthatóik megegyeznek
Polinom szorzás ellenőrzésének problémája: $P_1(x), P_2(x), P_3(x) \in \mathbf{F}[x]$, ellenőrizzük: $P_1(x) \times P_2(x) = P_3(x)$. Ha a szorzótényezők legfeljebb n-ed fokúak, akkor az eredmény polinom foka legfeljebb 2n. A szorzás ideje: O(n logn) Gyors Fourier Transzformációt használva (FFT), amikor egy pontban a kiértékelés ideje O(n).
$\mathbf{S} \subseteq \mathbf{F}$ mérete legyen legalább 2n+1. Vegyük $r \in \mathbf{S}$ –t véletlenszerűen egyenletes eloszlás alapján és értékeljük ki $P_1(r), P_2(r), P_3(r)$-t O(n) idő alatt. Az algoritmus csak akkor hibázik, ha a polinomok azonossága nem teljesül, de az algoritmus r esetén ezt nem ismeri fel.
Legyen $Q(x)= P_1(x) \times P_2(x) - P_3(x)$, 2n-es fokkal. Egy polinom azonosan 0, ha minden együtthatója 0, tehát Q(x) nál ez csak akkor teljesül, ha a szorzás helyes. De ha Q(x) nem azonosan 0, akkor nagy valószínűséggel $Q(r) = P_1(r) \times P_2(r) - P_3(r)$ sem 0.
Q-nak legfeljebb 2n különböző gyöke van. Tehát ha Q nem azonosan 0, akkor nem több mint 2n darab különböző r választásával Q(r)=0. Tehát a hiba maximum 2n / **|S|.** Ez csökkenthető két módon is: 1) az egész algoritmussal független iterációkat végzünk, 2) eléggé nagy **S** halmazt választunk.
Ha **F** végtelen test, akkor a hibahatár 0-ra csökkenthető, mivel r-t **F**-ből választjuk, csak ekkor végtelen számú random bitre van szükség. Ennek egy determinisztikus változatánál minden r-t , $r \in \mathbf{S}$, csak egyszer próbálunk ki. De ekkor minden polinom esetében 2n+1 különböző kiértékelésre van szükség, aminek a futási ideje $O(n \log^2 n)$ ami több mint ami $P_1(x) \times P_2(x)$-hez kell.

Valamikor az együtthatókkal való számolás túl drága, például mátrix determinánsa. Legyen **M** egy n x n-es mátrix. Ennek determinánsa:

$$\det(\mathbf{M})= \sum_{\pi \in |S_n} \operatorname{sgn}(\pi) \prod_{i=1}^{n} M_{i,\pi(i)}$$

Ahol **S** az n-méretű permutációk csoportja, $\operatorname{sgn}(\pi)$: π permutáció előjele. $\operatorname{sgn}(\pi)=(-1)^t$, ahol a t az egységpermutáció π-be alakításához szükséges páronkénti elemváltozások száma. A determinánsnak n! tagja van, de polinomiális időben kiértékelhető, ha $\mathbf{M}_{ij}$ értékeit explicite megadjuk.

Def: Vandermonde mátrix $M(x_1,..,x_n)$ a határozatlan $x_1,\ldots,x_n$ -ben kifejezve definiált, úgy mint $\mathbf{M}_{ij}= x_i^{j-1}$:

$$M = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ & & & \cdot & \\ & & & \cdot & \\ & & & \cdot & \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{pmatrix}$$

A Vandermonde azonosság erre a mátrixra: $\det(\mathbf{M})=\prod_{j<i} (x_i - x_j)$.

Mivel elég drága a szimbolikus mátrixok effajta determinánsának ellenőrzése, ezért a fenti módszert vesszük elő: $Q(x_1,\dots,x_n) = \det(\mathbf{M}) - \prod_{j<i} (x_i - x_j)$ azonosan 0 kell legyen.

Minden véletlen x_i-re tudjuk ellenőrizni, hogy Q azonosan 0-e. Q foka, a tagok fokainak maximuma lesz.

Schwartz-Zippel Tétel: Legyen $Q(x_1,\dots,x_n) \in \mathbf{F}[x_1,\dots,x_n]$ d fokú többváltozós polinom. Legyen $\mathbf{S} \subseteq \mathbf{F}$ egy véges halmaz, és legyen $r_1,\dots,r_n$ egyenletes eloszlással **S**-ből választva:

$$\mathbf{Pr}[Q(r_1,\dots,r_n)] = 0 \mid Q(x_1,\dots,x_n) \text{ nem azonosan } 0] \le d / |\mathbf{S}|.$$

Bizonyítás: A bizonyítás n változónak a számán alapuló teljes indukció.
n=1 : d fokú, egyváltozós polinomot ad: $Q(x_1)$, és az előzőekből láttuk, hogy $Q(x_1)$ nem azonosan 0, akkor annak a valószínűsége, hogy $Q(r_1)=0$, legfeljebb $d / |\mathbf{S}|$.
Tegyük fel hogy többváltozós polinomokra (n-1 változóig, n>1) az indukciós feltevés igaz.

$$Q(x_1,\dots,x_n) = \sum_{i=0}^{k} x_1^i Q_i(x_2,\dots,x_n), \text{ ahol}$$

$k \le d$ a legnagyobb kitevője x_1-nek Q-ban, mivel x_1 Q-ra hat, ezért $k>0$.
x_1^k együtthatója, $Q_k(x_2,\dots,x_n)$, nem azonosan 0, k választása alapján. Q_k teljes foka maximum d – k, az indukciós hipotézis alapján annak valószínűsége, hogy $Q_k(r_2,\dots,r_n)=0$, legfeljebb $(d – k) / |\mathbf{S}|$.
Feltehető, hogy $Q_k(r_2,\dots,r_n) \neq 0$, ekkor:

$$q(x_1)=Q(x_1, r_2, r_3,\dots, r_n) = \sum_{i=0}^{k} x_1^i Q_i(r_2,\dots,r_n).$$

Ez a polinom k-ad fokú, és nem azonosan 0, mivel az x_1^k együtthatója $Q_k(r_2,\dots,r_n)$. Az alapesetből következik, hogy annak a valószínűsége, hogy $q(r_1) = Q(r_1,\dots,r_n)$ kiértékelve 0, legfeljebb $k / |\mathbf{S}|$.

Így:

$$\mathbf{Pr}[Q_k(r_2,\dots,r_n)=0] \le (d – k) / |\mathbf{S}|$$
$$\mathbf{Pr}[Q(r_1,\dots,r_n)=0 \mid Q_k(r_2,\dots,r_n)\neq 0] \le k / |\mathbf{S}|$$

Annak a valószínűsége tehát, hogy $Q(r_1,\dots,r_n)=0$, nem több mint ezen két valószínűség összege, ami: $d / |\mathbf{S}|$. A bizonyítás kész.

Készítette: Sánta Róbert