Bonyolultságelmélet

Wednesday 2nd November, 2016, 21:14

Max-2SAT

A probléma

Input: egy $2CNF$, klózonként pontosan két literállal; **a literálok változói különböznek.**
Output: egyszerre legfeljebb hány klóz elégíthető ki benne?

Bonyolultság

Tudjuk, hogy a $2\mathrm{SAT}$ probléma $\mathbf{P}$-beli.
Viszont, ennek az optimalizálási problémának az eldöntési változata (input egy F $2CNF$ és egy K szám, kielégíthető-e F-ben egyszerre K klóz?) $\mathbf{NP}$-teljes.

Először adunk egy **randomizált $\frac{4}{3}$-közelítő algoritmust**, aztán ezt **derandomizálva** egy determinisztikusat.

Randomizált algoritmus

Random: várható értékben approximáljon

Egy A randomizált algoritmus α-approximáló, ha **várható értéke** legfeljebb α-szor rosszabb, mint az optimális, vagyis:

$$\forall x \ \max\left\{\frac{E(A(x))}{f(x)}, \frac{f(x)}{E(A(x))}\right\} \ \leq \ \alpha.$$

Max-2SAT

Ebben az esetben tehát egy olyan algoritmust kell adnunk, mely legalább $\frac{3}{4}X$ klózt kielégít az input formulában, ha legfeljebb X elégíthető ki egyszerre.
Ennél többet teszünk: olyan algoritmust fogunk adni, mely (várható értékben) a klózok $\frac{3}{4}$-ét (tehát nem csak az egyszerre kielégíthető klózokét!) igazzá teszi.

Randomizált algoritmus

Állítsunk minden változót egymástól függetlenül $\frac{1}{2} - \frac{1}{2}$ valószínűséggel igazra vagy hamisra.

Várható érték

Mivel egy klózban két különböző változó van, így annak esélye, hogy a klóz igaz lesz, $\frac{3}{4}$.
A várható érték additivitása miatt így várható értékben a klózok $\frac{3}{4}$-ét kielégítjük.

Derandomizálás

Most az előző dián szereplő véletlen algoritmusban csökkentjük a generálandó véletlen bitek számát: **derandomizálunk**.

Determinisztikus algoritmus

Legyenek $x_1, x_2, \ldots, x_n$ az input formulában szereplő változók.

- Először értéket adunk x_1-nek, majd x_2-nek, ..., végül x_n-nek, az i-edik menetben x_i-nek.
- Az i-edik menetben $x_1, \ldots, x_{i-1}$ értéke már rögzítve van.
- Számítsuk ki $b = 0, 1$-re, hogy mennyi klóz elégülne ki várható értékben, ha x_i értékét b-re választjuk, $x_{i+1}, \ldots, x_n$ értékét pedig $\frac{1}{2} - \frac{1}{2}$ valószínűséggel, függetlenül választjuk 0-nak ill. 1-nek.
- Amelyik érték nagyobb lett, arra állítjuk ténylegesen x_i értékét, és folytatjuk a ciklust.

Derandomizálás

Vegyük észre, hogy a várható értéket determinisztikusan ki tudjuk számítani.
Annak esélye, hogy egy klóz értéke igaz, a következőképp számítható:

- ha a klózban van olyan literál, melynek értékét 1-re rögzítettük, akkor 1;
- különben $1 - \frac{1}{2^k}$, ahol $0 \leq k \leq 2$ a klózban levő, még nem rögzített értékű változók száma.

A formulában igazra állított klózok értéke ezek összege.
Azt is be lehet könnyen látni, hogy minden menetben a várható érték nem csökkenhet, így az utolsó menet végére a ,,várható érték" továbbra is legalább a klózok $\frac{3}{4}$-e lesz.

Max-2SAT példa

Példa

$$F = (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_1 \vee \neg x_3) \wedge (x_2 \vee \neg x_3) \wedge (x_2 \vee x_4) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_2 \vee \neg x_3) \wedge (x_3 \vee x_4) \wedge (\neg x_3 \vee \neg x_4)$$

Determinisztikus

Ha $x_1 = 0$, a várható értékek: $\frac{1}{2}$, $\frac{1}{2}$, 1, 1, 1, 1, a többi $\frac{3}{4}$, összesen 9.5;
ha $x_1 = 1$, a várható értékek 1, 1, $\frac{1}{2}$, $\frac{1}{2}$, $\frac{1}{2}$, $\frac{1}{2}$, a többi $\frac{3}{4}$, összesen 8.5;
így legyen $x_1 = 0$.
Ezek után ha $x_1 = 0$ és $x_2 = 0$, a várható értékek 0, 1, 1, 1, 1, 1, $\frac{1}{2}$, $\frac{1}{2}$, 1, 1, $\frac{3}{4}$ és $\frac{3}{4}$, összesen 9.5;
ha pedig $x_1 = 0$ és $x_2 = 1$, a várható értékek 1, 0, 1, 1, 1, 1, 1, 1, $\frac{1}{2}$, $\frac{1}{2}$, $\frac{3}{4}$ és $\frac{3}{4}$, szintén 9.5. Legyen mondjuk $x_2 = 1$.
. . . és így tovább. . .

Töredékes Hátizsák

Algoritmus a Töredékes Hátizsákra

Erre a változatra a **mohó** algoritmus optimális:

- rendezzük csökkenőbe a tárgyakat $\frac{c_i}{w_i}$ szerint;
- eszerint a sorrend szerint vegyünk be annyi tárgyat teljesen, amennyit csak tudunk;
- az első tárgyat, ami nem fér be, pedig ,,törjük" úgy, hogy a töredék megtöltse a hátizsák maradék kapacitását.

Approximálás?

Adódik a következő ötlet a Hátizsák problémára:

- rendezzük csökkenőbe a tárgyakat $\frac{c_i}{w_i}$ szerint;
- eszerint a sorrend szerint vegyünk be annyi tárgyat, amennyit csak tudunk.

Ez az algoritmus **nem** α-approximálja a Hátizsák problémát, semmilyen α konstansra.

Példa

Ha pl. két tárgyunk van, $w_1 = 1$, $c_1 = 1$, $w_2 = W$ és $c_2 = W - 1$:

- az első tárgy fajlagosan értékesebb, mint a második
- a töredékes változat optimumhelye $(1, \frac{W-1}{W})$, értéke $1 + \frac{(W-1)^2}{W}$
- a mohó algoritmus lefele kerekítő változata tehát az $(1, 0)$ helyen felvett 1 értéket adja vissza
- az optimumhely $(0, 1)$, értéke $W - 1$.

Ha tehát W elég nagy, $1 \cdot \alpha < W - 1$ lesz, így az algoritmus ezen változata nem α-approximáló.

Kis módosítás

A következő algoritmus viszont 2-approximálja a Hátizsák problémát:

- Tegyük a tárgyakat fajlagos érték szerint csökkenő sorrendbe: $\frac{c_1}{w_1} \geq \frac{c_2}{w_2} \geq \ldots \geq \frac{c_n}{w_n}$.
- Számítsuk ki a Töredékes Hátizsák optimumhelyét: $(1, 1, \ldots, 1, x, 0, 0, \ldots, 0)$, ahol x a k. tárgy (amelyiket el kellett törnünk, mert már nem fért be).
- Adjuk vissza vagy $\sum_{i=1}^{k-1} c_i$-t, vagy c_k-t, amelyik nagyobb.

Tehát: vagy fajlagosan csökkenő sorrendben beteszünk, amit csak lehet, vagy az így éppen kimaradó tárgyat egyedül rakjuk be, amelyik jobb.

Hátizsák 2-approximálható

Amiért ez legalább az optimum felét adja:

- a töredékes hátizsák optimuma ennek a két értéknek az összegénél kisebb, így a kettő közül a nagyobb legalább a töredékes optimumának fele;
- a töredékes hátizsák optimuma (mivel relaxált változat) legalább akkora, mint a $0/1$ hátizsáké.

Pár kimaradt részlet

Az algoritmus így akkor nem működik, ha minden tárgy befér egyszerre (ekkor az lesz az optimumhely), vagy ha a k. tárgy egyedül se férne be (de az ilyen tárgyakat eleve nem kell felvegyük a listába), de ezek könnyen javíthatók.

Hátizsák $1+\epsilon$-approximálható

Nézzünk egy másik kézenfekvő algoritmust a Hátizsák probléma közelítő megoldására:

Skálázás ϵ-ra

- Legyen $w_1, \ldots, w_n, c_1, \ldots, c_n, W$ a Hátizsák probléma egy inputja és $\epsilon > 0$ egy valós szám.
- Oldjuk meg a $w_1, \ldots, w_n, c'_1, \ldots, c'_n, W$ feladatot dinamikus programozással, ahol $c'_i = \lfloor \frac{c_i}{c_{\max}} \cdot \lfloor \frac{n}{\epsilon} \rfloor \rfloor$, $c_{\max} = \max_i c_i$ és adjuk vissza ennek az eredményét.

Megjegyzés

Tehát mintha a maximális hasznú tárgy haszna $\lfloor \frac{n}{\epsilon} \rfloor$ lenne, a többi haszna pedig evvel arányosan skálázódna (egészrészt véve).

Hátizsák $1+\epsilon$-approximálható

A dinamikus algoritmus

A rekurzív összefüggés most:

$$T[i,c] = \begin{cases} 0 & \text{ha } c \leq 0; \\ \infty & \text{ha } 0 = i < c; \\ \min\{T[i-1,c], T[i-1,c-c_i]+w_i\} & \text{egyébként.} \end{cases}$$

„Mekkora zsák kell legalább, hogy az első i tárgyból meglegyen legalább c haszon?”

Időigény

Ennek az algoritmusnak az időigénye
$O(\text{poly}(n, n \max c, \log W)) = O(\text{poly}(n \max c \log W))$.
(Pszeudopolinomiális.)

Hátizsák $1+\epsilon$-approximálható

$O(\text{poly}(n \max c \log W))$

Ha $\max c = \lfloor \frac{n}{\epsilon} \rfloor$, akkor ez polinom időigény az eredeti input méretében.
Persze nem mindig szolgáltat optimális eredményt, az osztás utáni egészrész vételekor vesztett pontosság miatt.
De ha rögzítjük ϵ-t, akkor **polinom** időigényű és
$1+\epsilon$-approximáló algoritmus!

PTAS, FPTAS

PTAS

Azt mondjuk, hogy az f optimalizálási problémára van **polinomidejű approximációs séma**, avagy PTAS, ha van olyan algoritmus, melynek f inputja és egy $\epsilon > 0$ szám a bemenete és tetszőleges rögzített ϵ-ra $(1+\epsilon)$-approximáló polinomidejű algoritmus.

FPTAS

Azt mondjuk, hogy az f optimalizálási problémára van **teljesen polinomidejű approximációs séma**, avagy FPTAS, ha van rá olyan PTAS, mely tetszőleges (x, ϵ) inputra $\mathrm{poly}(|x|, \frac{1}{\epsilon})$ időben fut.

Az előző dián pl. egy FPTAS-t láttunk a HÁTIZSÁK problémára.

Ha egy problémára van FPTAS, azzal tetszőleges inputra tetszőleges pontossággal polinomidőben meg tudjuk határozni az optimumot.

FPTAS és pszeudopolinomiális algoritmusok

Tétel

Ha egy egészértékű optimalizálási probléma

i) optimuma korlátozható az inputméret egy polinomjával

ii) és van rá FPTAS,

akkor van rá pszeudopolinomiális algoritmus.

Ötlet

Ha a fenti korlát n^c, akkor $\epsilon = \frac{1}{n^c}$ megfelelő választás lesz.

Következmény

Ha $\mathbf{P} \neq \mathbf{NP}$ és egy, a fenti i) tulajdonsággal rendelkező probléma eldöntési változata erősen $\mathbf{NP}$-teljes, akkor nem lehet rá FPTAS.

Néhány további eredmény a $\mathbf{P}$ és $\mathbf{NP}$ osztályokról

Eddig minden $\mathbf{NP}$-beli problémánkról vagy azt mutattuk meg, hogy $\mathbf{P}$-ben van, vagy azt, hogy $\mathbf{NP}$-teljes.
Azonban...

Tétel (Ladner, 1975)

Ha $\mathbf{P} \neq \mathbf{NP}$, akkor létezik olyan $L \in \mathbf{NP} - \mathbf{P}$, mely nem $\mathbf{NP}$-teljes.

$\mathbf{P}$ és $\mathbf{NPC}$ közt?

Gráf-izomorfizmus

Input: két gráf.
Output: izomorfak-e?

Faktorizálás

Input: $N, K > 0$ egészek.
Output: Van-e N-nek K-nál kisebb prímosztója?

A fenti két probléma egyikéről sem ismert, hogy $\mathbf{P}$-beliek-e, sem az, hogy $\mathbf{NP}$-teljesek lennének.

Megjegyzés

A Faktorizálásra persze van pszeudopolinomiális algoritmus, tehát ha $\mathbf{P} \neq \mathbf{NP}$, akkor nem lehet **erősen** $\mathbf{NP}$-teljes; másfelől co$\mathbf{NP}$-beli is, így ha $\mathbf{NP} \neq \text{co}\mathbf{NP}$, nem lehet $\mathbf{NP}$-teljes.

A co**NP** osztály

Definíció

co$\mathbf{NP} = \{L : \overline{L} \in \mathbf{NP}\}$

Példák

Érvényesség

Adott: φ Boole-formula

Kérdés: Érvényes-e, azaz azonosan igaz-e φ?

Hamilton-út Komplementer

Adott: G gráf

Kérdés: Igaz-e, hogy G nem tartalmaz Hamilton-utat?

Állítás

Az Érvényesség és Hamilton-út Komplementer problémák co**NP**-ben vannak.

NP és coNP

Egy probléma akkor van

- **NP**-ben, ha minden igen példányához létezik polinom méretű, polinom időben ellenőrizhető bizonyíték, és csak az igen példányokhoz létezik ilyen bizonyíték.
- co**NP**-ben, ha minden nem példányhoz létezik polinom méretű, polinom időben ellenőrizhető cáfolat, és csak a nem példányokhoz létezik ilyen cáfolat.

Állítás

Az Érvényesség és Hamilton-út Komplementer problémák co**NP**-teljesek.

NP és coNP

Állítás

$$\mathbf{P} \subseteq \mathbf{NP} \cap \mathrm{co}\mathbf{NP}.$$

Állítás

Tegyük fel, hogy $\mathcal{C}$ zárt a visszavezetésre és L $\mathcal{C}$-teljes.
Ekkor $\mathcal{C} = \mathrm{co}\mathcal{C} \Leftrightarrow L \in \mathrm{co}\mathcal{C}$.

Következmény

$$\mathbf{NP} = \mathrm{co}\mathbf{NP} \;\Leftrightarrow\; \textsc{Sat} \in \mathrm{co}\mathbf{NP} \;\Leftrightarrow\; \textsc{Érvényesség} \in \mathbf{NP}.$$

Determinisztikus algoritmusokkal...

- 3SAT eldönthető $\tilde{\mathcal{O}}(1.3303^n)$ időben (Makino, Tamaki, Yamamoto, 2011),
- 3-SZÍNEZÉS eldönthető $\mathcal{O}(1.3289^n)$ időben (Beigel, Eppstein, 2005),
- FÜGGETLEN CSÚCSHALMAZ eldönthető $\mathcal{O}(1.2108^n)$ időben (Robson, 1986 – Fomin, Grandoni, Kratsch 2009),
- ...

Az **Exponenciális Időhipotézis** azt a sejtést fogalmazza meg, hogy a 3SAT-ot (és sok más **NP**-teljes problémát) nem lehet **szubexponenciális** időben megoldani.

Ez a sejtés erősebb, mint a $\mathbf{P} \neq \mathbf{NP}$...

Összefoglalás

- Adtunk randomizált $4/3$-közelítő algoritmust a MAX-2SATra.
- Derandomizáltuk az előző algoritmust, így kaptunk egy determinisztikus $4/3$-approximálást a MAX-2SATra.
- Megadtunk egy egyszerű algoritmust, mely 2-approximálja a HÁTIZSÁKot.
- Láttuk, hogy a HÁTIZSÁK probléma $1+\epsilon$-approximálható tetszőleges ϵ-ra.
- Definiáltuk a polinomidejű approximációs sémákat (PTAS) és a teljesen polinomidejűeket (FPTAS).
- Láttunk egy kapcsolatot teljesen polinomidejű approximációs sémák és pszeudopolinomiális algoritmusok közt.
- Kimondtuk Ladner tételét.
- Megismertük a sokak által **NP**-köztesnek sejtett GRÁF-IZOMORFIZMUS és FAKTORIZÁLÁS problémákat.
- Megismertük az exponenciális időhipotézist.