Übersicht

Einführung

Grundlagen

Methoden zum Entwurf von Approximationsalgorithmen

Approximationsklassen

Das PCP-Theorem

Probabilistische Algorithmen

- Verbesserter Algorithmus für MAXIMUM SAT
- Semidefinite Programmierung
- Derandomisierung

Algorithmus für Vertex Cover

Minimum Weighted Vertex Cover

Instanz:	ungerichteter Graph $G = (V, E)$, Gewicht $w(v)$ für $v \in V$.
Lösung:	vertex cover $U \subseteq V$
Maß:	$\sum_{u \in U} w(u)$

$U := \emptyset$
while $E \neq \emptyset$ do
  wähle $e = \{u, v\} \in E$
  wähle $x \in \{u, v\}$ zufällig mit $\mathbb{P}(x = u) = \frac{w(v)}{(w(u)+w(v))}$
  $U := U \cup \{x\}$
  $E := E \setminus \{e\,;\, x \in e\}$
return U

Satz

Für das Maß m_{Pr} dieses vertex cover U gilt $E[m_{Pr}] \leq 2m^*$

Einfacher Algorithmus

SimpleSAT(F)

für jede Variable x

setze $\alpha(x) = 1$ oder $\alpha(x) = 0$,

jeweils mit Wahrscheinlichkeit $\frac{1}{2}$

return α

Satz

Für eine Instanz F von Maximum SAT

mit $|C| \geq k$ für jede Klausel $C \in F$ gilt:

$$E[m(F, \alpha)] \geq \left(1 - \frac{1}{2^k}\right) m$$

Lineares Programm für SAT

Sei $F = C_1 \wedge \ldots C_m$ in Variablen $x_1, \ldots, x_n$.

Ganzzahliges lineares Programm $IP(F)$:

Variablen $z_1, \ldots, z_m$ und $y_1, \ldots, y_n$

$$\begin{array}{ll} \text{maximize} & \sum_{j=1}^{m} z_j \\ \text{subject to} & \displaystyle\sum_{x_i \in C_j} y_j + \sum_{\bar{x}_i \in C_j} (1 - y_i) \geq z_j \quad \text{für } j = 1, \ldots, m \\ & y_i \in \{0, 1\} \\ & z_j \in \{0, 1\} \end{array}$$

Rationale Lockerung: $LP(F)$

Zufälliges Runden

Algorithmus für MAXIMUM SAT:

LP-SAT(F)

Berechne optimale Lösung (y^*, z^*) für $LP(F)$
Für $i = 1, \ldots, n$ setze
$\alpha(x_i) = 1$ mit Wahrscheinlichkeit y_i^*
$\alpha(x_i) = 0$ mit Wahrscheinlichkeit $1 - y_i^*$

Satz

Sei $a_k := 1 - (1 - 1/k)^k$.
Für eine Formel F *mit* $|C| \leq k$ *für jede Klausel* $C \in F$ *gilt:*

$$E[m(F, \alpha)] \geq a_k m^*(F)$$

Lemma

Für $j = 1, \ldots, m$ *mit* $|C_j| = k$ *gilt:*

$$1 - \prod_{x_i \in C_j} (1 - y_i^*) \prod_{\bar{x}_i \in C_j} y_i^* \ \geq \ a_k z_j^*$$

Kombinierter Algorithmus

$KombiSAT(F)$

$\alpha_1 := SimpleSAT(F)$
$\alpha_2 := LP\text{-}SAT(F)$
if $m(F,\alpha_1) \geq m(F,\alpha_2)$
then return α_1
else return α_2

Satz

Für jede Instanz F von MAXIMUM SAT *gilt:*

Ist $\alpha := KombiSAT(F)$, so ist

$$E[m(F,\alpha)] \geq \frac{3}{4} m^*(F) \, .$$

Symmetrische Matrizen

$A \in \mathbb{R}^{n \times n}$ ist symmetrisch $(A \in SYM_n)$, wenn

$A^T = A$ ist, also $a_{i,j} = a_{j,i}$ für alle i, j.

Skalarprodukt von Matrizen $A, B \in SYM_n$:

$$A \bullet B := \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} b_{i,j} \qquad = \mathrm{Tr}(A^T B)$$

Frobenius-Norm: $\quad ||A|| := \sqrt{A \bullet A} = \sqrt{\sum_{i,j} a_{i,j}^2}$

Eigenschaft:

Ist $A \in SYM_n$, so sind alle Eigenwerte λ von A reell, $\lambda \in \mathbb{R}$

Positiv semidefinite Matrizen

$A \in \mathbb{R}^{n \times n}$ ist positiv semidefinit ($A \succeq 0$), wenn

- A ist symmetrisch: $\quad A \in SYM_n$
- für alle Eigenwerte λ von A: $\quad \lambda \geq 0$.

Satz

Sei $A \in SYM_n$. Dann sind äquivalent:

- $A \succeq 0$
- $x^T A x \geq 0 \quad$ *für alle* $x \in \mathbb{R}^n$.
- *es gibt* $U \in \mathbb{R}^{n \times n}$ *mit* $A = U^T U$.

Cholesky-Faktorisierung

Der folgende Algorithmus berechnet für $M \succeq 0$ ein U mit $M = U^T U$ in $O(n^3)$ Operationen:

Cholesky(M)

if dim $M = 1$

let $M = (a)$

return $(\sqrt{a})$

else

let $M = \begin{pmatrix} a & q^T \\ q & N \end{pmatrix}$

$V := \mathit{Cholesky}\left(N - \frac{1}{a} q q^T\right)$

return $\begin{pmatrix} \sqrt{a} & \frac{1}{\sqrt{a}} q^T \\ 0 & V \end{pmatrix}$

Semidefinite Programmierung

Verallgemeinerung von linearer Programmierung:

- Statt $x \in \mathbb{R}^n$ suche Lösung $X \in SYM_n$.
- Bedingung $x \geq 0$ wird verallgemeinert zu $X \succeq 0$

Maximierungsproblem Semidefinite Programming

Instanz:	Matrizen $A_1, \ldots, A_m, C \in \mathbb{R}^{n \times n}$, Vektor $b \in \mathbb{R}^m$,
Lösung:	Matrix $X \in SYM_n$ mit $X \succeq 0$ und $A_k \bullet X = b_k$ für $k = 1, \ldots, m$
Maß:	$C \bullet X$

Lösung semidefiniter Programme

Ellipsoid-Methode gibt Algorithmus E mit:

Sei $P = (A_1, \ldots, A_m, C, b)$ ein semidefinites Programm, mit $\|X\| \leq r$ für alle $X \in S(P)$.

Für jedes $\epsilon > 0$ findet E eine Lösung $X \in S(P)$, mit

$$m(P, X) \geq m^*(P) - \epsilon$$

in Zeit polynomiell in $|P|$ und $\log(r/\epsilon)$.

Das Problem Maximum Cut

Problem Maximum Cut

Instanz:	Graph $G = (V, E)$
Lösung:	Partition $V = V_0 \cup V_1$ mit $V_0 \cap V_1 = \emptyset$
Maß:	Anzahl der Kanten $\{x, y\}$ mit $x \in V_0$ und $y \in V_1$.

Probabilistischer Algorithmus:

für jedes $v \in V$
  wähle $b \in \{0, 1\}$ zufällig
  $V_b := V_b \cup \{v\}$

Satz

Der Algorithmus liefert einen Lösung mit Maß m_R, so dass $E[m_R] \geq \frac{1}{2} m^*$.

Von Maximum Cut zum Vektorprogramm

Maximum Cut als ganzzahliges quadratisches Programm:

$$\begin{array}{ll} \text{maximize} & \frac{1}{2} \sum_{\{v_i, v_j\} \in E} (1 - y_i y_j) \\ \text{subject to} & y_i^2 = 1 \\ & y_i \in \mathbb{Z} \end{array}$$

Relaxation zum Vektorprogramm $Vek(G)$:

$$\begin{array}{ll} \text{maximize} & \frac{1}{2} \sum_{\{v_i, v_j\} \in E} (1 - z_i^T z_j) \\ \text{subject to} & ||z_i|| = 1 \\ & z_i \in \mathbb{R}^n \end{array}$$

Es gilt: $m^*(Vek(G)) \geq m^*(G)$

Von Maximum Cut zum semidefiniten Programm

Daraus erhält man das semidefinite Programm $SDP(G)$:

$$\begin{array}{ll} \text{maximize} & \displaystyle\frac{1}{2}\sum_{\{v_i,v_j\}\in E}(1-x_{i,j}) \\ \text{subject to} & x_{i,i}=1 \\ & (x_{i,j})=X\succeq 0 \\ & X\in\mathbb{R}^{n\times n} \end{array}$$

Satz

Vektorprogramm $Vek(G)$ und semidefinites Programm $SDP(G)$ sind äquivalent, insb. gilt $m^*(SDP(G))=m^*(Vek(G))$.

Algorithmus von Goemans-Williamson

- berechne $Vek(G)$ und $SDP(G)$
- finde Lösung X von $SDP(G)$ mit $m(X) \geq m^*(SDP(G)) - \epsilon$
- daraus Lösung $z_1, \ldots, z_n$ von $Vek(G)$ mit $m(z_1, \ldots, z_n) \geq m^*(Vek(G)) - \epsilon$
- wähle zufällig $p \in \mathbb{R}^n$ mit $||p|| = 1$.
- definiere $V_0 = \{x_i\,;\, p^T z_i \geq 0\}$ und $V_1 = \{x_i\,;\, p^T z_i < 0\}$

$m_{GW}(G)$ ist die Größe des so berechneten Schnittes.

Analyse von Goemans-Williamson

Satz

Es gilt: $E[m_{GW}(G)] \geq 0,87856 \cdot m^*(G)$

Lemma

Für alle $-1 \leq z \leq 1$ *gilt:*

$$\frac{1}{\pi}\arccos(z) \geq 0,878567\frac{1-z}{2}$$

Quadratisches Programm für Maximum 2-SAT

Gegeben 2-KNF $F = C_1 \wedge \ldots \wedge C_m$ in Variablen $x_1, \ldots, x_n$.

Variablen y_i für $i = 0, \ldots, n$ mit Werten $y_i \in \{-1, 1\}$

Idee: $x_i = 1 \quad \leadsto \quad y_i = y_0$, also $y_0 y_i = 1$

$x_i = 0 \quad \leadsto \quad y_i \neq y_0$, also $y_0 y_i = -1$

- $v(x_i) = (1 + y_0 y_i)/2$
- $v(\bar{x}_i) = (1 - y_0 y_i)/2$
- $v(a \vee b) = 1 - v(\bar{a}) v(\bar{b})$

Beispiel: $v(x_i \vee \bar{x}_j) = \frac{1}{4}(1 + y_0 y_i) + \frac{1}{4}(1 - y_0 y_j) + \frac{1}{4}(1 + y_i y_j)$

Vektorprogramm für Maximum 2-SAT

Quadratisches Programm:

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{m} v(C_i) = \sum_{0 \le i < j \le n} a_{i,j}(1 + y_i y_j) + b_{i,j}(1 - y_i y_j) \\
\text{subject to} \quad & y_i^2 = 1 \\
& y_i \in \mathbb{Z}
\end{aligned}$$

Relaxation zum Vektorprogramm $Vek(F)$:

$$\begin{aligned}
\text{maximize} \quad & \sum_{0 \le i < j \le n} a_{i,j}(1 + z_i^T z_j) + b_{i,j}(1 - z_i^T z_j) \\
\text{subject to} \quad & ||z_i|| = 1 \\
& z_i \in \mathbb{R}^{n+1}
\end{aligned}$$

Algorithmus für MAXIMUM 2-SAT

- berechne $Vek(F)$
- finde Lösung $z_1, \ldots, z_n$ von $Vek(F)$ mit $m(z_1, \ldots, z_n) \geq m^*(Vek(F)) - \epsilon$
- wähle zufällig $p \in \mathbb{R}^n$ mit $||p|| = 1$.
- definiere $y_i = \begin{cases} 1 & p^T z_i \geq 0 \\ -1 & p^T z_i < 0 \end{cases}$
- definiere $\alpha(x_i) = \begin{cases} 1 & y_i = y_0 \\ 0 & y_i \neq y_0 \end{cases}$

Satz

Für das Maß $m_{SD}(F)$ der berechneten Bewertung α gilt:

$$E[m_{SD}(F)] \geq 0,87856 \cdot m^*(F)$$

Methode der bedingten Erwartung

Algorithmus A wählt zufällige Werte $v_1, v_2, \ldots$

$\rightsquigarrow$ Berechnungsbaum für Instanz x

- jeder Knoten $\triangleq$ Folge $v_1 v_2 \ldots v_i$ von gewählten Werten.
- jedes Blatt b $\triangleq$ vollständige Berechnung $\rightsquigarrow$ Wert $m_{A,b}(x)$

Für Knoten u im Berechnungsbaum:

$E(v)$: Mittelwert der Werte $m_{A,b}(x)$ für Blätter b unter u.

Wurzel r: $\quad E(r) = E[m_A(x)]$

Methode der bedingten Erwartung

ObdA sei A Algorithmus für Maximierungsproblem.

Wenn für jeden Knoten u mit Kindern u_0, u_1

$E(u_0) \geq E(u_1)$ effizient entscheidbar

$\rightsquigarrow$ deterministischer Algorithmus A' mit $m_{A'}(x) \geq E[m_A(x)]$

A': wähle bei jedem Knoten u den Wert für u_i
so dass $E(u_i)$ maximal.

Dann gilt $E(u_i) \geq E(u)$.

Also bei erreichtem Blatt b:

$$m_{A'}(x) = m_{A,b}(x) \geq \ldots \geq E(r) = E[m_A(x)]$$

Derandomisierung von *SimpleSAT*

$m_S(F)$: Wert von *SimpleSAT* bei F

Zu entscheiden:

$$E[\, m_S(F) \,|\, v_1 \ldots v_i\, 0\,] \geq E[\, m_S(F) \,|\, v_1 \ldots v_i\, 1\,]$$

Der Wert $E[\, m_S(F) \,|\, v_1 \ldots v_i\, 1\,]$ (und analog $E[\, m_S(F) \,|\, v_1 \ldots v_i\, 1\,]$ kann in polynomieller Zeit berechnet werden.

Damit: deterministischer Algorithmus A mit $m_A(F) \geq \frac{1}{2} m^*(F)$.