Approximation Algorithms

Maurice QUEYRANNE

Lecture 5
August 7, 2009

Recall **Weighted MAXCUT** [Vazirani, Chap. 26]

given a graph $G=(V,E)$ with edge weights $w_{uv} > 0 \ \forall uv \in E$

find $S \subseteq V$ which maximizes $f(S) = \sum_{\substack{uv\in E \\ u\in S, v\notin S}} w_{uv} = W(S, V\setminus S)$

A quadratic IP formulation: let $y_v = \begin{cases} +1 & \text{if we choose } v \in S \\ -1 & \text{o/w } (v \in V\setminus S)\end{cases}$ (n variables)

QIP $\left[\begin{array}{lll} \max & \frac{1}{2}\sum_{i<j} w_{ij}(1-y_iy_j) & \\ \text{s.t.} & y_i^2 = 1 \quad (y \text{ integer}) & \forall i \in V \end{array}\right.$

define $y_{ij} = y_i y_j \quad \forall i,j$ (n^2 variables)

i.e., an $n\times n$ matrix Y which should be $= y\,y^T$

$\Rightarrow Y$ should be symmetric

with all diagonal terms $y_{ii} = 1$

and psd - because $\forall x \in \mathbb{R}^n \quad x^TYx = x^T(yy^T)x = (x^Ty)(y^Tx)$

$= (x^Ty)^2 \geq 0$

our SDP relaxation only uses these 3 properties:

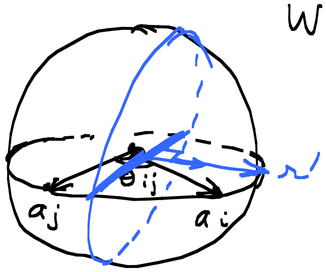
SDP-MC $\left[\begin{array}{ll} \max \frac{1}{2}\sum_{i<j} w_{ij}(y_{ii}-y_{ij}) & = C\cdot Y \text{ for some matrix } C \\ \text{s.t.} \quad y_{ii} = 1 \ \forall i & A_i \bullet Y = 1 \text{ for a matrix } A_i \ \forall i \\ Y \succcurlyeq 0, \quad Y \in M_n & \end{array}\right.$

Let $Y = (a_1, \ldots, a_n)$ be an ~~(approximate)~~ optimum solution to SDP-MC

$y_{ii} = 1 \iff a_i^Ta_i = 1$, i.e., each $a_i \in S_{n-1}$ unit sphere in $\mathbb{R}^n$

Let θ_{ij} be the angle between a_i and a_j

Their contribution to the objective is $\frac{1}{2}w_{ij}(1-\cos\theta_{ij})$



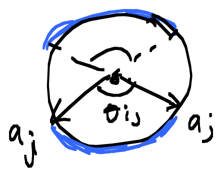
We will construct a cut $(S, V\setminus S)$ by separating these vectors $a_1 \dots a_n$ into two subsets using a **hyperplane** $H := \{x \in \mathbb{R}^n : r^T x = 0\}$ through the origin

$S := \{a_i : r^T a_i \le 0\}$ $\quad V\setminus S = \{a_i : r^T a_i > 0\}$

with **normal vector r** uniformly distributed on S_{n-1}

Lemma: $\text{Prob}\{r^T a_i \text{ and } r^T a_j \text{ are of different signs}\} = \dfrac{\theta_{ij}}{\pi}$

proof: in the 2-dimensional subspace (plane) generated by a_i and a_j, H divides the unit disk into two halves, with a uniformly distributed random normal vector r' in this plane. QED



Remark: we can generate a uniformly distributed random vector r on the unit sphere S_{n-1} as follows

- let $x_1 \dots x_n$ denote a random sample from the standard Normal distribution $N(0,1)$
- let $d := \sqrt{x_1^2 + \dots + x_n^2}$ and all components $r_j = x_j / d$

(optional exercise: verify that the resulting r is uniformly distributed on S_{n-1})

Goemans & Williamson's MaxCut algorithm

1- find a (near) optimum solution $Y = (a_1 \dots a_n)$ to SDP-MC

2. draw a vector r uniformly on the unit sphere S_{n-1}

3 let $S = \{i : r^T a_i \ge 0\}$

Let $W := f(S)$ the weight of this cut (a random variable)

$$\alpha := \frac{2}{\pi} \min_{0 \le \theta \le \pi} \frac{\theta}{1-\cos\theta} > 0.87856\ldots$$

Lemma: $E[W] \ge \alpha \, OPT_{SDP\text{-}MC}$

proof: for any $\theta \in [0, \pi]$ $\quad \dfrac{\theta}{\pi} \ge \alpha \dfrac{1-\cos\theta}{2}$

by the previous Lemma $E[W] = \sum_{i<j} w_{ij} \, \text{Prob}\{i \in S \text{ and } j \notin S\} = \sum_{i<j} w_{ij} \dfrac{\theta_{ij}}{\pi}$

$$\ge \alpha \sum_{i<j} w_{ij} (1 - \cos\theta_{ij}) = \alpha \, OPT_{SDP\text{-}MC}$$

QED

<u>Theorem</u>: There exists a randomized $\frac{1}{\alpha}$-approximation for MAX-CUT where $\frac{1}{\alpha} \approx 1.138$

proof. (sketch) repeat the GW MAX-CUT algorithm a polynomial number of times to get this approximation guarantee with high probability QED

Approximation Schemes [Vazirani, Chap. 8]

a family of ε-approximate algorithms for a given problem, for all $\varepsilon > 0$

i.e., an algorithm which depends on ε and produce a solution S_ε to a combinatorial optimization problem $\max\{f(S) : S \in \mathcal{S}\}$ such that $f(S_\varepsilon) \geq (1-\varepsilon)\,OPT$

A Polynomial Time Approximation Scheme (PTAS) is an approximation scheme with running time polynomial in the input size, for every $\varepsilon > 0$

A Fully Polynomial Time Approximation Scheme (FPTAS) is a PTAS with running time polynomial in the input size and in $1/\varepsilon$

0-1 Knapsack problem

given a set N of n objects with volume $vol(i) > 0$ and integer profit $prof(i) > 0$ and a capacity $B > 0$ where each $vol(i) \leq B$

find a set of items with total volume $\leq B$ and largest total profit

IP formulation: let $x_j := \begin{cases} 1 & \text{if we choose item } j \\ 0 & \text{o/w} \end{cases}$

$$\left[\begin{array}{l} \max \sum_j prof(j)\, x_j \\ \text{s.t. } \sum_j vol(j)\, x_j \leq B, \quad x \in \mathbb{B}^n \end{array}\right.$$

A pseudopolynomial time algorithm is an algorithm whose running time is polynomial in the total value of all input numbers

– recall that the input size of a number B is $\lceil \log_2 B \rceil + 1$

Dynamic Programming Algorithm for the Knapsack problem:

let $P := \max\{prof(i) : i \in N\}$ so $OPT \le nP$

for every $i \in \{1..n\}$ and $p \in \{0, 1, .., nP\}$ let S_{ip} be a subset of $\{1, .., i\}$ with total profit exactly p and minim total volume

and $A(i,p)$ to be its total volume

($+\infty$ if no S_{ip} exists)

DP algorithm:

initialization: $A(0,p) := \begin{cases} 0 & \text{if } p = 0 \\ +\infty & \text{o/w} \end{cases}$

recursion: for $i = 1, \dots n$

for $p = 0 \dots nP$ do

$A(i,p) = \min\{A(i-1,p),\ vol(i) + A(i-1, p - prof(i))\}$

record which alternative is chosen

$OPT = \max\{p : A(n,p) \le B\}$

trace back an optim solution using the recorded choices

running time $O(n^2 P)$ pseudo polytime

FPTAS for Knapsack: use rounding of the item profits

1. Given $\varepsilon > 0$ let $K := \frac{\varepsilon P}{n}$
2. For every item i let $prof'(i) = \left\lfloor \frac{prof(i)}{K} \right\rfloor$
3. Run the DP algorithm on this rounded instance
4. Return the resulting (rounded instance) optimum S'

Remark: S' is feasible

$$P' = \max_i \text{prof}'(i) \le \max_i \frac{\text{prof}(i)}{\varepsilon P/n} = \frac{n}{\varepsilon}$$

running time $O(n^2 P') = O(n^3 \frac{1}{\varepsilon})$

Lemma: In the original problem $\text{prof}(S') \ge (1-\varepsilon)\,OPT$

proof: let O be an optimum solution to the original problem

since for each item i $\text{prof}'(i) \ge \frac{\text{prof}(i)}{K} - 1$

$$\text{prof}(i) \le K\,\text{prof}'(i) + K$$

$$\text{prof}(O) - K\,\text{prof}'(O) \le nK$$

O is a feasible solution to the rounded problem

$$\text{prof}(S') \ge K\,\text{prof}'(O) \ge \text{prof}(O) - nK = OPT - \varepsilon P \ge (1-\varepsilon)\,OPT$$

because $OPT \ge P$ (any single item defines a feasible solution) QED

$\Rightarrow$ Thm: This algorithm is an FPTAS for the Knapsack problem

Difficulty of Approximation [Vazirani, Chap. 29]

recall the inapproximability result for TSP (general nonnegative lengths)

decision problem: HAMILTONIAN CYCLE problem: given a graph $G=(V,E)$ does G contain a Hamiltonian cycle?

$\downarrow$

TSP instance: on complete graph (V, E') on V

with edge lengths $\ell_{ij} = \begin{cases} 0 & \text{if } ij \in E \\ 1 & \text{o/w.} \end{cases}$

- if G contains a Ham. cycle then $OPT = 0$
- else $OPT \ge 1$

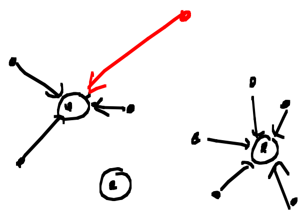
METRIC K-CENTER problem [Vazirani Chap. 5]

given a complete graph $G=(V,E)$ and edge lengths $\ell_{ij} \geq 0$ satisfying the triangle inequality, and an integer k

find a subset S of at most k vertices, which minimizes

$$\max_{v} \ \text{connect}(v,S)$$

where connect(v,S) is the minimum length of an edge connecting v to S



edge length = geometric distance

$k=3$

example: locate emergency facilities (firefighter stations, ambulance depots, ...) so that each demand point is within the smallest possible distance (response time) of its closest service facility

METRIC K-CENTER problem is NP-hard

DOMINATING SET problem:

given a graph $G=(U,F)$ find the smallest subset $T \subseteq U$ such that every $u \in U$ is either in T or adjacent to a vertex in T

DOMINATING SET is NP-hard

Thm: If $P \neq NP$ then for every $\varepsilon > 0$ there does not exist a $(2-\varepsilon)$-approximation for METRIC K-CENTER

proof: by contradiction, assume there exist such ε and $(2-\varepsilon)$ approximation

Given an instance $H=(U,F)$ of DOMINATING SET problem

For any integer $k \in \{1, \dots, |U|\}$

define an instance of METRIC K-CENTER where G is the complete graph over U, edge lengths $\ell_{ij} = \begin{cases} 1 & \text{if } uv \in F \\ 2 & \text{o/w} \end{cases}$

Note that ℓ satisfies the Δ-inequality

- if there exist a dominating set S in H with at most k vertices then use this set as solution to k-center instance so $\text{connect}(v, S) \leq 1 \quad \forall v \in U$ so $OPT_{MKC} = 1$
- else $\text{connect}(v, S) \geq 2$ for any subset S with $|S| \leq k$ and $OPT_{MKC} \geq 2$

With our $(2-\varepsilon)$-approximation algorithm we can decide whether H contains a dominating set of size $\leq k$.

Repeating this for all $k = 1, \ldots, |U|$ we can find the least feasible value of k, we can solve any DOMINATING SET instance in poly time. Therefore $P = NP$, a contradiction QED

The PCP Theorem and Inapproximability results

- The PCP Theorem provides an alternative characterization of NP, from which stronger inapproximability results can be derived
- there are other characterizations of NP that lead to even stronger inapproximability results
- the "Unique Games Conjecture" (UGC) about another characterization of NP leads, if it is true, to even stronger results

	Known approx.	Inapprox. if $P \neq NP$	Inapprox. if UGC
Vertex Cover	2	$10\sqrt{5} - 21 \approx 1.36$	$2 - \varepsilon$
MAX CUT	$1.138 + \varepsilon$	$17/16$	$1.138 - \varepsilon$
MAX 3SAT	$4/3$	$8/7$	
CLIQUE	(n)	$n^{1-\varepsilon}$ (*)	

(*) if $NP \neq ZPP$, the class of decision problem that admit admit a Las Vegas algorithm, a randomized algorithm which always gives a correct answer, with <u>average</u> running time polynomial in the input size

MAX 3SAT is the special case of MAXSAT with at most 3 literals per clause

CLIQUE: given a graph $G=(V,E)$ find a largest $S \subseteq V$ such that $\forall i,j \in S \quad ij \in E$

PCP Probabilistically Checkable Proof

given two functions $r(n)$ ("<u>r</u>andom bits")
$q(n)$ ("<u>q</u>ueries")

$PCP(r(n), q(n))$ is the set of languages (families of bit strings) that are accepted by a verifier, a special Turing machine V with 3 input tapes:
- for the instance (string) x
- for a string of random bits
- for the proof (or certificate), another string y

and which, for every word x, uses
$O(r(|x|))$ random bits
and $O(q(|x|))$ bits from the supplied proof y
such that
if $x \in L$ then there exists a proof y which V accepts with probability 1
else, then for <u>every</u> proof y, V accepts x with probability $< 1/2$

<u>PCP Theorem:</u> $NP = PCP(\log n, 1)$

Maximum Acceptance Probability (MAP) problem:

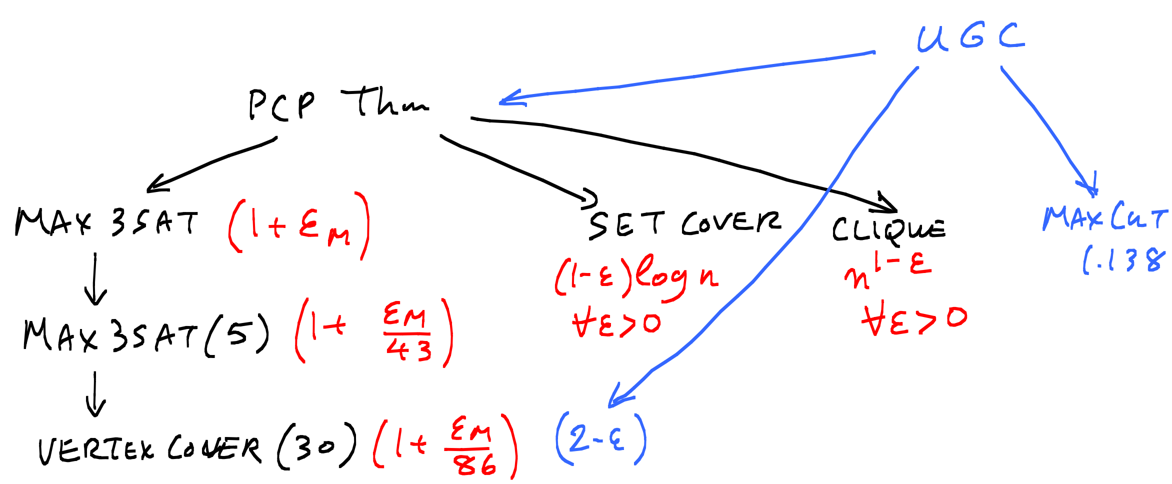
Let V be a PCP(log n, 1) verifier for SAT

Given any (unsatisfiable) logical expression ϕ (in CNF) find a proof y which maximizes the acceptance probability of ϕ by V

Thm: If there exists a ½-approximation for MAP then P=NP

the results in the second column of the above table are derived from this theorem using appropriate transformations

Some reductions (see Vazirani Chap. 29 for the reductions shown by black arrows)



MAX 3SAT(k) special case of MAX 3SAT in which each variable appears at most k times

ε_M is a specific positive constant (see Thm 29.7 in Vazirani)

VERTEX COVER(d) — - VERTEX COVER in which the maximum degree is ≤ 30

The UGC is another characterization which is conjectured for NP