Reading: Text: Williamson-Shmoys, Chapter 6

Semidefinite Programming: Max-Cut

Problem: MAX-CUT.

Given

- graph $G = (V, E)$
- weights $w : E \to \Re^+$

Output

- set S that maximizes $\sum_{i \in S, j \in \bar{S}} w_{ij}$

Claim: MAX-CUT is NP-hard.

Question: Algorithms?

- random cut, (1/2)-approx.
- local search, (1/2)-approx.
- linear programming

$$\begin{array}{ll} \max & \sum_{e \in E} w_e x_e \\ s.t. & x_{ij} + x_{jk} + x_{ik} \leq 2 \\ & x_{ij} + x_{jk} \geq x_{ik} \end{array}$$

Claim: No matter how you round this, it is at best a (1/2)-approx.

Proof: Find a graph with small max cut but large LP value, e.g., square plus diagonal.

[[*Can strengthen with more inequalities, but doesn't help in general.*]]

- semidefinite programming

SDP Formulation

Idea: Variables on vertices indicating side of cut.

Quadratic Integer Program

$$v_i = \begin{cases} 1 & : i \in S \\ -1 & : i \notin S \end{cases}$$

$$\begin{array}{ll} \max & \sum_{(i,j) \in E} \frac{1 - v_i v_j}{2} \\ s.t. & v_i \in \{-1, 1\} \end{array}$$

Quadratic Linear Program

Rewrite constraint as $v_i^2 = 1$ and relax v_i to be linear (i.e., $v_i \in \Re$).

[[*Exactly max-cut, so still not solvable.*]]

Semidefinite Program

Relax v_i to be real vector, $v_i \in \Re^n$:

$$\begin{array}{ll} \max & \sum_{(i,j)\in E} \frac{1 - v_i \cdot v_j}{2} \\ s.t. & v_i \cdot v_i = 1 \\ & v_i \in \Re^n \end{array}$$

Looks hard, but in fact solvable!

Idea: Just a slightly more general LP.

Let $\rho_{ij} = v_i \cdot v_j$:

$$\begin{array}{ll} \max & \sum_{(i,j)\in E} \frac{1 - \rho_{ij}}{2} \\ s.t. & \rho_{ij} = 1 \end{array}$$

Unbounded, must also constrain ρ_{ij} to be an inner product.

Let P be $n \times n$ matrix of inner products of v_i's. Then:

- P is symmetric since $\rho_{ij} = v_i \cdot v_j = v_j \cdot v_i = \rho_{ji}$, so all eigenvalues are real.
- All diagonals are 1 since $\rho_{ii} = v_i \cdot v_i = 1$.
- Let $V = [v_1\ v_2\ \ldots\ v_n]$. Then $P = V^T V$ (importantly, $\exists$ a matrix V s.t. $P = V^T V$), so for any $x \in \Re^n$,

  $$x^T P x = x^T V^T V x = (Vx)^T (Vx) = ||Vx||^2 \geq 0.$$

Def: Such a matrix P is called *positive semidefinite* (above 3 conditions equiv.).

Adding constraint $x^T P x \geq 0$ for all $x \in \Re^n$ gives an infinite LP:

$$\begin{array}{ll} \max & \sum_{(i,j)\in E} \frac{1 - \rho_{ij}}{2} \\ s.t. & \rho_{ij} = 1 \end{array}$$

$$\sum_{ij} x_i x_j \rho_{ij} \geq 0, \forall x \in \Re^n$$

Solve using ellipsoid:

Def: A *separation oracle* checks in polytime whether proposed solution P satisfies all constraints or else produces a constraint that is violated.

Fact: Ellipsoid solves LPs in polytime if there's a separation oracle.

Separation Oracle:

- Given P, check symmetric
- Compute eigenvalues
- If non-negative, done
- Else eigenvector x with negativie eigenvalue is violating constraint

Rounding the SDP

Goal: Want (i, j) to be cut when $(1-v_i \cdot v_j)/2$ is large.

Draw picture:

- unit vectors v_i are directions in sphere
- large dot product, directions nearly same
- small dot product, directions nearly opposite
- want to separate far apart vectors

Idea: Separating hyperplane.

Question: Which hyperplane?

Note: Opt vectors rotationally invariant, so doesn't make sense to bias toward some direction.

- Choose unit vector $r \in \Re^n$ uniformly
- Let $S = \{i : v_i \cdot r \geq 0\}$

Note: To chose random r, pick coordinates from IID from Gaussian and normalize.

Claim: (Goemans-Williamson): This is a γ-approx where $\gamma = \min_{-1 \leq x \leq 1} \frac{2 \cos^{-1} x}{\pi(1-x)} \approx 0.87856$.

Proof: By LOE, we have:

$$E[w(S : \bar{S})] = \sum_{ij} w_{ij} \Pr[(i,j) \in (S : \bar{S})].$$

Consider 2-dim. space spanned by v_i and v_j.

- Let p be projection of r
- Note $r \cdot v_i = p \cdot v_i$ and $r \cdot v_j = p \cdot v_j$
- Note p uniform over unit circle in plane of v_i and v_j as r is uniform over sphere

Therefore,

$$\Pr[(i,j) \in (S, \bar{S})] = \angle(v_i, v_j)/\pi.$$

But $v_i \cdot v_j = \cos(\angle(v_i, v_j))$!

$$\begin{aligned} E[w(S, \bar{S})] &= \sum_{i,j} w_{ij}(\angle(v_i, v_j)/\pi) \\ &= \arccos(v_i \cdot v_j)/\pi \end{aligned}$$

and SDP value is $\sum_{i,j} w_{ij}(1 - v_i \cdot v_j)/2$, so look for worst angle for ratio, i.e.,

$$\min_{-1 \leq x \leq 1} (\arccos(x)/\pi)/((1-x)/2)$$

[[*Analysis is tight and problem is NP-hard to approx. within 16/17. Assuming UGC, alg. is optimal.*]]

Semidefinite Programming: Correlation Clustering

Problem: Max Correlation Clustering:

Have input in which each pair of objects displays a degree of similarity or dissimilarity, want to cluster similar objects together, split dissimilar objects apart.

Given:

- Graph $G = (V, E)$
- Weights $w_{ij}^+ \geq 0$ and $w_{ij}^- \geq 0$ for each edge

Output:

- Clustering that maximizes sum of w^+ weights inside clusters plus w^- weights between clusters:

$$\sum_{(i,j) \in E(S)} w_{ij}^+ + \sum_{(i,j) \in \delta(S)} w_{ij}^-$$

Question: Approximations?

- better of all-together/all-apart is (1/2)-approx
- semidefinite programming

Let $v_i \in \Re^n$ be unit vector indicating cluster of vertex i:

$$\begin{aligned} \max \quad & \sum_{(i,j) \in E} \left(w_{ij}^+ (v_i \cdot v_j) + w_{ij}^- (1 - v_i \cdot v_j) \right) \\ s.t. \quad & v_i \in \{e_1, \ldots, e_n\} \end{aligned}$$

Relax to SDP:

$$\max \quad \sum_{(i,j) \in E} \left(w_{ij}^+ (v_i \cdot v_j) + w_{ij}^- (1 - v_i \cdot v_j) \right)$$

$$
\begin{aligned}
s.t. \quad & v_i \cdot v_i = 1 \\
& v_i \cdot v_j \geq 0 \\
& v_i \in \Re^n
\end{aligned}
$$

Rounding: choose 2 uniformly random vectors r_1 and r_2:

- $R_1 = \{i : r_1 \cdot v_i \geq 0, r_2 \cdot v_i \geq 0\}$
- $R_2 = \{i : r_1 \cdot v_i \geq 0, r_2 \cdot v_i < 0\}$
- $R_3 = \{i : r_1 \cdot v_i < 0, r_2 \cdot v_i \geq 0\}$
- $R_4 = \{i : r_1 \cdot v_i < 0, r_2 \cdot v_i < 0\}$

Claim: Above is a 0.75-approx.

Proof: Let X_{ij} indicate i and j are split. Then

$$E[X_{ij}] = (1 - \arccos(v_i \cdot v_j)/\pi)^2$$

Claim: For $x \in [0,1]$,

$$\frac{(1 - \arccos(x)/\pi)^2}{x} \geq 0.75$$

and

$$\frac{1 - (1 - \arccos(x)/\pi)^2}{x} \geq 0.75$$

Since $v_i \cdot v_j \in [0,1]$, above claim implies

$$E[W] \geq 0.75 \sum_{i,j} \left(w_{ij}^+ (v_i \cdot v_j) + w_{ij}^- (1 - v_i \cdot v_j) \right) \geq 0.75 \times \text{OPT}.$$

Note: Get (3/4)-approx with just 4 clusters!