Lecturer: Arnab Bhattacharyya

- Recall our SDP relaxation for MAX-2-SAT:

$$\max \sum_{1 \le i \le j \le n} a_{i,j}(1 - v_i \cdot v_j) + b_{i,j}(1 + v_i \cdot v_j)$$

$$\text{s.t. } \|v_i\| = 1.$$

GW rounding gives approx factor $0.878\cdots$

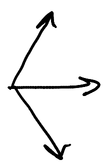
- Consider 2 SAT instance with the single clause $x_i \vee x_j$.

SDP is:

$$\max \frac{1 + v_0 \cdot v_i}{4} + \frac{1 + v_0 \cdot v_j}{4} + \frac{1 - v_i \cdot v_j}{4}$$

$$\text{s.t. } \|v_0\| = \|v_i\| = \|v_j\| = 1.$$

Feasible solution: $v_0 = (1, 0)$, $v_i = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$, $v_j = \left(\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)$



Value of SDP is: $\frac{3}{8} + \frac{3}{8} + \frac{3}{8} = \frac{9}{8}$

Integrality gap is $\le \frac{8}{9} = 0.888\cdots$

- In bad instance, clause is "over-satisfied"

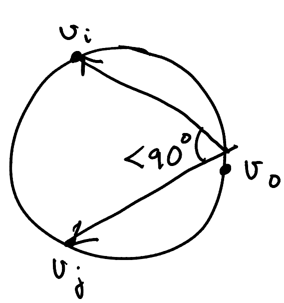
<u>Adding triangle constraints</u>

- Since clauses are never over-satisfied legitimately, we can explicitly add a constraint enforcing it.
- For any clause $x_i \vee x_j$, add the constraint:

$$\frac{1 + v_0 \cdot v_i}{4} + \frac{1 + v_0 \cdot v_j}{4} + \frac{1 - v_i \cdot v_j}{4} \le 1$$

$$\text{or, } 1 - v_0 \cdot v_i - v_0 \cdot v_j + v_i \cdot v_j \ge 0$$

$$\text{or, } (v_i - v_0) \cdot (v_j - v_0) \ge 0$$



So, $\|v_i - v_j\|^2 \le \|v_i - v_0\|^2 + \|v_j - v_0\|^2$

"Triangle inequality for ℓ_2^2 distance"

- Add such constraints for each clause $(x_i \vee x_j, x_i \vee \bar{x}_j, \bar{x}_i \vee x_j, \bar{x}_i \vee \bar{x}_j)$

- "Canonical" SDP relaxation
- [LLZ02]: Integrality gap, approx factor ≥ 0.940
- [Rag 08]: Assuming UGC, approx ratio = integrality gap.
- In some sense, the best possible relaxation because triangle inequality imply all other valid local constraints (explore in problem set).

Approximating MAX k-CSP's

- What are constraint satisfaction problems?
  - Domain D. We'll stick to $D = \{0,1\}$.
  - Integer $k > 0$.
  - List P of k-ary predicates, each mapping $D^k \to \{0,1\}$.

Instance of k-CSP[P] is a set of clauses

$$P_1(x_{i_{11}}, \dots, x_{i_{1k}}),\ P_2(x_{i_{21}}, \dots, x_{i_{2k}}), \dots, P_m(x_{i_{m1}}, \dots, x_{i_{mk}})$$

where $P_1, \dots, P_m \in P$ and $i_{11}, \dots, i_{mk} \in \{1, \dots, n\}$.

MAX-k-CSP[P]: find an assignment $x_i \to a_i$ which satisfies as many clauses as possible.

- Examples: MAX-CUT, k-LIN, MAX-k-SAT, ...
- Let's look at MAX-3-SAT (setup will generalize to other MAX k-CSP[P] problems)
- For i'th variable of SAT instance, have a variable t_i. In integer solution, $t_i = 1$ would mean i'th variable set to TRUE. Also, want $t_i(1 - t_i) = 0$ so that t_i would be $\{0,1\}$-valued.
- In vector version, replace 1 with some vector e. We have the constraint $t_i \cdot (e - t_i) = 0$. We interpret

$e \cdot t_i$ as "prob. of setting i'th var to TRUE"

and $t_i \cdot t_j$ as "prob. of setting both i and j'th vars to TRUE"

- Again, we can do MAX-2-SAT easily

$$x_i \vee x_j \longrightarrow t_i \cdot t_j + t_i \cdot (e - t_j) + (e - t_i) \cdot t_j$$

with constraints $t_i \cdot (e - t_i) = 0$ and $e^T e = 1$

- Can easily check this is equivalent to our earlier formulation but no triangle inequality and unclear how to generalize to MAX-3-SAT with SDP.

- Idea: Introduce new variables
  For clause j, have 8 variables $z_{j,TTT}, z_{j,FTT}, \ldots, z_{j,FFF}$
  intuitively meaning that if $z_{j,TFT} = 1$ and clause j has the variables x_3, x_5, x_6, then $x_3 = \text{TRUE}$, $x_5 = \text{FALSE}$, $x_6 = \text{TRUE}$.

New constraints: $\forall\, j \in [m]$, $\sum_{\omega \in \{0,1\}^3} z_{j,\omega} = 1$.

- We interpret $z_{j,\omega}$ as "prob. that assignment on scope of clause j is ω".

z_j is a prob. distribution on "local assignments" to clause j.

- But how to relate the z variables with the t variables?

We do the best we can using SDP constraints.

- Marginal on one variable

For clause j and $i \in [k]$, let $v_i(j)$ = id of i'th var in clause j.
E.g. if clause j is $x_2 \vee \overline{x}_5 \vee x_6$, $v_1(j) = 2$, $v_2(j) = 5$, $v_3(j) = 6$.

- New constraints:

For each $j \in [m]$, $i \in [k]$:

$$\sum_{\substack{\omega \in \{F,T\}^3 : \\ \omega_i = T}} z_{j,\omega} = e \cdot t_{v_i(j)}$$

Marginal on two variables

New constraints:

For each $j \in [m]$, $i, i' \in [k]$:

$$\sum z_{j,\omega} = t_{v_i(j)} \cdot t_{v_{i'}(j)}$$

$\overleftarrow{\omega \in \{F,T\}^3:}$
$\omega_i = \omega_{i'} = T$

In summary, canonical SDP relaxation of MAX-3-SAT:

$$\max \sum_{j=1}^{m} \sum_{\substack{\omega \in \{F,T\}^3: \\ C_j \text{ sat by } \omega}} z_{j,\omega}$$

$$\begin{aligned}
\text{s.t.} \quad & t_i \cdot (e - t_i) = 0 && \forall\, i \in [n] \\
& e \cdot e = 1 && \\
& z_{j,\omega} \geq 0 && \forall\, j \in [m],\ \omega \in \{F,T\}^3 \\
& \sum_{\omega} z_{j,\omega} = 1 && \forall\, j \in [m] \\
(*) \quad & \sum_{\omega:\, \omega_i = T} z_{j,\omega} = e \cdot t_{v_i(j)} && \forall\, j \in [m],\ i \in [k] \\
(**) \quad & \sum_{\substack{\omega:\, \omega_i = T, \\ \omega_{i'} = T}} z_{j,\omega} = t_{v_i(j)} \cdot t_{v_{i'}(j)} && \forall\, j \in [m],\ i, i' \in [k]
\end{aligned}$$

Note: $(**) \Rightarrow (*)$ by setting $i = i'$.

<u>Why canonical?</u>

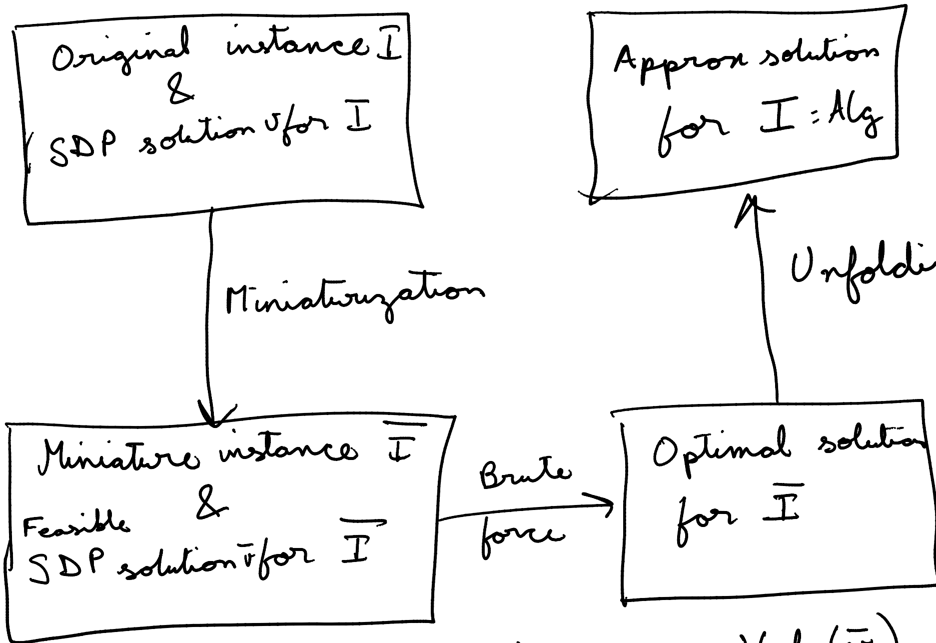
- That it's a relaxation is clear.
- Can show that feasible solutions to canonical SDP satisfies triangle inequality and any other "local" SDP constraint satisfied by legit solutions.
- Can view the t_i's as generating real random variables J_i s.t. $\mathbb{E}[J_i J_j] = t_i \cdot t_j$.
- In the ideal case, each J_i would be a constant, either 1 or 0, indicating whether var i should be TRUE or FALSE.

<u>Why?</u> <u>Fact</u>: A matrix M is psd iff $\exists$ real random vars $X_1, \ldots, X_n$ s.t. $M_{ij} = \mathbb{E}[X_i X_j]$.

- Not ideal, but can verify that $\mathbb{E}[J_i^2] = \mathbb{E}[J_i]$.

Rounding algorithm for canonical SDP

- Algorithm and guarantee very similar to MAX-CUT algorithms discussed in the last class.



$$
\begin{aligned}
Alg = Opt(\bar{I}) &\geq Gap \cdot SDP(\bar{I}) \geq Gap \cdot Val(\bar{v}) \\
&\geq Gap \cdot (SDP(v) - \varepsilon m) \\
&\geq Gap \cdot (OPT - \varepsilon m) \\
&\geq OPT \cdot Gap \cdot (1 - O(\varepsilon)).
\end{aligned}
$$

- Algorithm :
  - Suppose $v = (e, t_1, \ldots, t_n, z_{j,\omega} : j \in [m], \omega \in \{F, T\}^3)$.
  - Apply dimension reduction by a random Gaussian projection $\phi = \underbrace{\left[\quad\right]}_{n} \updownarrow d$
  - Let $e^* = \phi(e)$, $t_i^* = \phi(t_i)$
  - Repeat until $\|e^*\| \in [1-\delta, 1+\delta]$ and for $< \varepsilon m$ clauses, $\|e^* \cdot t_i^* - e \cdot t_i\| > \delta$ or $\|t_i^* \cdot t_j^* - t_i \cdot t_j\| > \delta$ for i and j occurring in the clause.
  - Discard failed clauses and obtain instance I^*.

- Discretize by moving each t_i^* to nearest point of ε-net with k points
- Fold the formula as before to get ~~weighted~~ #MAX 3-SAT instance on k variables
- Solve by brute force and unfold.

- Need to show that an "almost feasible" solution can be fixed to a truly feasible solution.
  - Will not go through details of this part but it's constructive.