Goemans and Williamson (1995) — "Goemans and Williamson Algorithm for MAXCUT and Gronthendieck's Inequality"

January 29, 2018; rev. February 8, 2018

Kevan Hollbach, Lily Li

1 Semi-Definite Programming Review

2 Goemans Williamson Algorithm (Standard SDP)

Let $G = (V, E)$ be a weighted undirected graph with $|V| = n$ and where each edge (i, j) has weight $w_{ij} \geq 0$. The goal of the MAXCUT problem is to find a partition $(S, V - S)$ which maximizes the sum of the edge weights of edges crossing the partition.

We can formulate the problem as the following quadratic program:

$$\begin{aligned} \text{Maximize:} \quad & \sum_{(i,j)\in E} \frac{w_{ij}(1 - x_i x_j)}{2} \\ \text{Subject to:} \quad & x_i \in \{-1, +1\}, \text{ for } i \in [n] \end{aligned}$$

where x_i is associated with vertex v_i and $x_i x_j = 1$ if and only if v_i and v_j are placed in the same set. Let OPT denote the optimum solution to this quadratic program.

Next we introduce the vector programming relaxation of the above quadratic program:

$$\begin{aligned} \text{Maximize:} \quad & \sum_{(i,j)\in E} \frac{w_{ij}(1 - \mathbf{u_i} \cdot \mathbf{u_j})}{2} \\ \text{Subject to:} \quad & \|\mathbf{u}_i\|^2 = 1 \text{ and } \mathbf{u}_i \in \mathbb{R}^n, \text{ for } i \in [n]. \end{aligned}$$

To see that this is indeed a relaxation, take $\mathbf{u}_i = (\underbrace{x_i, 0, ..., 0}_{n})$ for each $i \in [n]$. Observe these $\mathbf{u}_i$'s satisfy the constraints ($\|\mathbf{u}_i\|^2 = 1$ and $\mathbf{u}_i \in \mathbb{R}^n$) and $\mathbf{u_i} \cdot \mathbf{u_j} = x_i x_j$. Thus, if OPT_{VP} denotes the optimum solution to this vector program then $OPT_{VP} \geq OPT$.

The above vector program is equivalent to the following semidefinite program:

$$\begin{aligned} \text{Maximize:} \quad & \sum_{(i,j)\in E} \frac{w_{ij}(1 - x_{ij})}{2} \\ \text{Subject to:} \quad & x_{i,i} = 1 \text{ for } i \in [n] \text{ and } X \succeq 0 \end{aligned}$$

where X has entry $x_i j$ in row i column j (two see that these two forms are equivalent remark that $X \succeq 0$ if and only if $X = U^T U$). Solve the semi-definite program in polynomial time and obtain some optimal solution X^*.

Cholesky factorize X^* into $(U^*)^T U^*$ and let the columns of U^*, $\mathbf{u}_i^* \in \mathbb{R}^n$, be the solutions to the vector program. We want to round each $\mathbf{u}_i^*$ to some $x_i^* \in \{-1, +1\}$. Then $x_i = x_i^*$ will be a solution to our original quadratic program. Do this in the following randomized manner: pick $\mathbf{r} = (r_1, ..., r_n)$ by drawing each r_i independently from the distribution $\mathcal{N}(0, 1)$. Then let

$$x_i^* = \begin{cases} 1 & \mathbf{u}_i^* \cdot \mathbf{r} \geq 0 \\ -1 & \text{otherwise} \end{cases}.$$

It is helpful to have the geometric picture in mind: each $\mathbf{u}_i^*$ is a vector which lies on the n-dimensional unit sphere. The hyper-plane with normal $\mathbf{r}$ split the sphere in-half. All $\mathbf{u}_i^*$ in the same half of the sphere gets mapped to the same value $c \in \{-1, 1\}$ and all the $\mathbf{u}_j^*$ in the other half gets mapped to $-c$.

To show the constant of approximation we need to consider the probability that an edge e_{ij} gets cut. This is equivalent to the probability that $\mathbf{u}_i$ and $\mathbf{u}_j$ fall in different halves of the sphere cut by the hyper-plane normal to $\mathbf{r}$. Consider the projecting of the normalized vector $\mathbf{r}$ onto the span of $\{\mathbf{u}_i, \mathbf{u}_j\}$. See Figure 1.

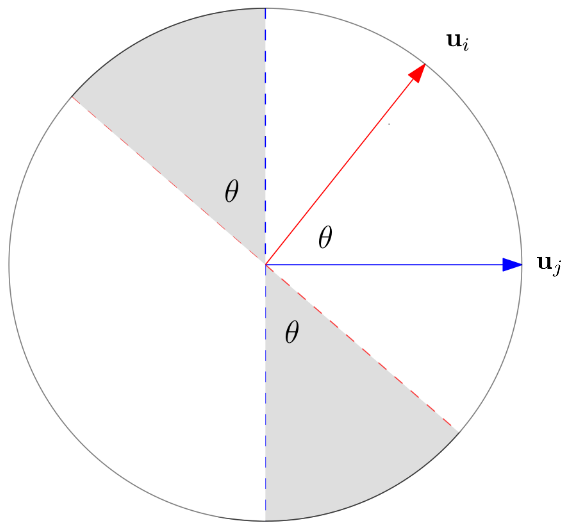

Figure 1: Shaded part denote regions where the normalized $\mathbf{r}$ can lie such that $\mathbf{u}_i \cdot \mathbf{r}$ and $\mathbf{u}_i \cdot \mathbf{r}$ have different sign.

Thus the probability that $\mathbf{r} \cdot \mathbf{u}_i$ and $\mathbf{r} \cdot \mathbf{u}_j$ have different sign is $\frac{\theta}{\pi}$. Since $\theta = \arccos(\mathbf{u}_i \cdot \mathbf{u}_j)$,

$$\Pr[e_{ij} \text{ is in the cut}] = \frac{\arccos(\mathbf{u}_i \cdot \mathbf{u}_j)}{\pi}. \tag{1}$$

We state without proof that

$$\frac{\arccos(x)}{\pi} \geq 0.878 \frac{1-x}{2} \tag{2}$$

for $x \in [-1, 1]$ — it helps to observe that the constant approximately minimizes $f(x) = \frac{2\arccos(x)}{\pi(1-x)}$ on the interval $[-1, 1]$. Thus the expected sum of weights obtained by the algorithm is

$$\begin{aligned} \mathbb{E}[W] &= \sum_{(i,j)\in E} w_{ij} \Pr[(i,j) \text{ is in the cut}] \\ &= \sum_{(i,j)\in E} w_{ij} \frac{\arccos(\mathbf{u}_i \cdot \mathbf{u}_j)}{\pi} && \text{by 1} \\ &\geq 0.878 \cdot \left(\sum_{(i,j)\in E} w_{ij} \frac{1-x_{ij}}{2} \right) && \text{by 2} \\ &= 0.878 \cdot OPT_{VP} \geq 0.878 \cdot OPT \end{aligned}$$

Since we can find a cut of size $0.878 \cdot OPT_{VP}$, $OPT \geq 0.878 \cdot OPT_{VP} \geq 0.878 \cdot OPT$.

3 SOS (Lassarre Hierarchy) Review

4 Pseudo-Distribution

Definition 1. *Let $f : \{0,1\}^n \to \mathbb{R}$. The **formal expectation** of f with respect to another function μ (not necessarily a probability distribution since it could be negative is*

$$\tilde{\mathbb{E}}_\mu f = \sum_{x \in \{0,1\}^n} \mu(x) \cdot f(x)$$

*A **degree-d pseudo-distribution** over $\{0,1\}^n$ is a function $\mu : \{0,1\}^n \to \mathbb{R}$ such that every formal expectation with respect to μ satisfies $\tilde{\mathbb{E}}_\mu 1 = 1$ and for every polynomial f of degree at most $\frac{d}{2}$*

$$\tilde{E}_\mu f^2 \geq 0.$$

Let a formal expectation with respect to a pseudo distribution μ of degree d is a degree d pseudo-expectation.

5 Vector Representation

Let y be a feasible solution in $\text{SOS}_t(K)$. We can equivalently represent y as vectors $\{v_I\}$, $|I| \leq t$, such that $y_{I \cup J} = \langle v_I, v_J \rangle$ for all $|I|, |J| \leq t$. This representation arises from the Cholesky decomposition of the moment matrix, $M_t(y)$, into

6 Goemans Williamson Algorithm (SOS $t = 3$)

We are now ready to present the GW algorithm through the lens of a level three SOS program. Let our graph $G = (V, E)$ be as the above with $|V| = n$ and where each edge (i,j) has weight $w_{ij} \geq 0$. Formulate MAXCUT as the following integer linear program:

$$\begin{aligned}
\text{Maximize:} \quad & \sum_{(i,j) \in E} w_{ij} z_{ij} \\
\text{Subject to:} \quad & \max(x_i - x_j, x_j - x_i) \leq z_{ij} \leq \min(x_i + x_j, 2 - x_i - x_j) \text{ for } (i,j) \in E, \\
& x_i, z_{i,j} \in \{0,1\} \text{ for } i \in [n] \text{ and } (i,j) \in E
\end{aligned}$$

where x_i is the indicator variable for a vertex chosen to be in set S of the partition and z_{ij} is the indicator variable for an edge crossing the cut $(S, V - S)$. Observe that $z_{ij} = |x_i - x_j|$.

Let K be the feasible region of the LP relaxation of the above integer program. For any graph we can set $x_i = \frac{1}{2}$ and $z_{i,j} = 1$ in the LP and obtain a value of $\sum_{(i,j) \in E} w_{ij}$. In particular, since the max cut of a complete graph on n-vertices with unit weight edges is at most $\frac{|E|}{2}$ while the output to the relaxation is $|E|$, the integrality gap of this LP scheme is 2.

Suppose instead that we started with $\mathbf{y} \in SOS_3(K)$. Pay particular attention to the elements of $\mathbf{y}$ indexed by the degree one monomials x_i and $z_{i,j}$, and the multinomials $x_i x_j$ for all $(i,j) \in E$. Denoted these as y_i, ζ_{ij}, and y_{ij} respectively. $\mathbf{y}$ has the form

$$\mathbf{y} = [y_\emptyset, y_1, ..., y_n, \zeta_{i,j}, ..., y_{i,j}, ...]$$

Lemma 2. *For any edge $(i,j) \in E$, $\zeta_{i,j} = y_i + y_j - 2y_{i,j}$.*

Proof. □

Using the vector representation (add reference), there exists vectors $\mathbf{v}_i$ such that $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = y_{i,j}$ for all $i, j \in [n]$. Recall however that the angle between any two vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ is between 0 and $\frac{\pi}{2}$ so applying hyper-plane rounding on the $\mathbf{v}_i$'s would be sub-optimal (we want the vectors to be more spread out so that a random hyper-plane would be more likely to separate a pair of verticies belonging to different sets).

Perform the vector transformation $\mathbf{u}_i = 2\mathbf{v}_i - \mathbf{v}_\emptyset$. Observe that $\mathbf{u}_i$ is a unit vector on the sphere centered at the origin. See Figure. In essence this transformation takes $[0,1]^n$ vectors $\mathbf{v}_i$ to $[-1,1]^n$ vectors $\mathbf{u}_i$ before rounding $\mathbf{u}_i$ to ± 1.

Lemma 3. *The vectors $\mathbf{u}_i = 2\mathbf{v}_i - \mathbf{v}_\emptyset$ form a solution to the vector program (add reference to the above) where $\zeta_{i,j} = \frac{1-\mathbf{u}_i \cdot \mathbf{u}_j}{2}$.*

Proof. We need to show that $\mathbf{u}_i$ is a unit vector and that $z_{i,j} = \frac{1-\mathbf{u}_i \cdot \mathbf{u}_j}{2}$ holds. Observe that

$$\mathbf{u}_i^2 = (2\mathbf{v}_i - \mathbf{v}_\emptyset)^2 = 4\mathbf{v}_i^2 - 4\mathbf{v}_i\mathbf{v}_\emptyset + \mathbf{v}_\emptyset^2 = 1$$

since $\langle \mathbf{v}_i, \mathbf{v}_i \rangle = y_i = \langle \mathbf{v}_i, \mathbf{v}_\emptyset \rangle$ and $\mathbf{v}_\emptyset^2 = y_\emptyset = 1$. Further

$$\frac{1 - \mathbf{u}_i \cdot \mathbf{u}_j}{2} = \frac{1 - (2\mathbf{v}_i - \mathbf{v}_\emptyset) \cdot (2\mathbf{v}_j - \mathbf{v}_\emptyset)}{2} = \mathbf{v}_i \cdot \mathbf{v}_\emptyset + \mathbf{v}_j\mathbf{v}_\emptyset - 2\mathbf{v}_j \cdot \mathbf{v}_i = y_i + y_j + 2y_{i,j} = \zeta_{i,j}.$$

□

7 Grothendieck's Inequality

Reference

Goemans, M. X. and Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145.