

Lecture 3: Lasserre Hierarchy - Properties and Applications

Professor Moses Charikar

Scribe: Neha Gupta

Overview We present some properties of the *Lasserre* hierarchy and then present some applications. A good reference is Rothvoss' lecture notes [Rot13].

1 Geometric properties of Lasserre solutions

Given two sets S_1, S_2 such that $|S_1|, |S_2| \leq t$, similarly to Sherali-Adams, the solutions of Lasserre hierarchy for t -levels induces a collection of probability distributions over subsets of at most t -variables. These distributions are also locally consistent, i.e. agree on the intersection $S_1 \cap S_2$. Moreover, as we will see next, the solutions of Lasserre have extra geometric structure that cannot be captured by a family of locally consistent distributions as in Sherali-Adams.

If $y \in \text{Las}_t(K)$, then we know that $\forall |I|, |J| \leq t$ there exist vectors v_I, v_J satisfying:

$$v_I \cdot v_J = y_{I \cup J} \quad (1)$$

$$\|v_I\|^2 = y_I \quad (2)$$

$$\|v_\phi\|^2 = 1 \quad (3)$$

For a variable i , we have the following set of constraints

$$v_i \cdot v_i = y_i \quad (4)$$

$$v_i \cdot v_\phi = y_i \quad (5)$$

These vectors v_i lie on a sphere of radius $\frac{1}{2}$ and center v_ϕ i.e. each v_i can be written as the following:

$$v_i = \frac{1}{2}(v_\phi + z) \text{ where } \|z\|_2^2 = 1 \quad (6)$$

We will verify one direction in the class and the reader can verify the other direction. Assume $v_i = \frac{1}{2}(v_\phi + z)$,

$$v_i \cdot v_\phi = \frac{1}{2}(1 + v_\phi \cdot z) \quad (7)$$

$$v_i \cdot v_i = \frac{1}{4}(1 + 1 + 2v_\phi \cdot z) = \frac{1}{2}(1 + v_\phi \cdot z) \quad (8)$$

This holds more generally. For any sets S and T such that $S \subseteq T$ and $|T| \leq t$, we have that:

$$v_S \cdot v_T = v_T \cdot v_T \quad (9)$$

$$v_T \text{ lies on a sphere of radius } \frac{1}{2}\sqrt{y_S} \text{ and center } \frac{1}{2}v_S \quad (10)$$

2 Knapsack problem

In the Knapsack problem, we are given n items each with a certain weight w_i and value v_i as well as a knapsack of total weight W . Our goal is to find a set of elements S to pack into the knapsack such that their total weight $w(S) = \sum_{i \in S} w_i \leq W$ does not exceed W and the total value of the items $v(S) = \sum_{i \in S} v_i$ is maximized.

Now, consider this instance of the knapsack problem, where $v_i = w_i = 1$ for all $i \in [n]$ and $W = 2 - \epsilon$ for $\epsilon > 0$. In this case, we know that the optimum value for this instance is 1 and we are interested to see how different relaxations of the problem behave.

2.1 LP relaxation

We consider the natural LP relaxation for this problem. Let x_i denote whether the item i is picked or not.

$$\begin{aligned} \max \quad & \sum_{i=1}^n x_i \\ \text{s.t} \quad & \sum_{i=1}^n x_i \leq 2 - \epsilon \\ & 0 \leq x_i \leq 1 \quad \forall i \in [n] \end{aligned}$$

It is easy to see that this LP relaxation is bad as for $x = \frac{2 - \epsilon}{n} \mathbf{1}$ it has a value of $2 - \epsilon$ whereas the value in any valid integral solution is at most 1. Thus, it is natural to ask of how many rounds it would take for a hierarchy like Sherali-Adams or Lasserre to discover that the optimum is 1.

2.2 Sherali-Adams Hierarchy

We will think about how many rounds would it take for Sherali Adams hierarchy to discover that the optimum is 1. Let us consider 2 rounds of Sherali Adams relaxation. Remember that Sherali-Adams maintains values y_S for all $|S| \leq 2$.

For this, we know that $y_{ij} = 0$ otherwise we could condition on i being 1 and get non zero probability of j being 1 which is not a feasible solution. It is not very hard to show that Sherali-Adams “cheats”, i.e. outputs solutions with value larger than 1, even for any number of rounds less than n .

Exercise 3.1. For Sherali Adams with 2 rounds, how high can $\sum_{i=1}^n y_i$ be for valid solutions $y \in \text{SA}_2(LP)$.

2.3 Lasserre Hierarchy

Next, we will see that the extra geometric structure of Lasserre solutions is able to overcome this problem within 2 rounds. Let $y \in \text{Las}_2(K)$, we know that the moment matrix $M_2(y) \succeq 0$ and that there exists a vector v_i for every item i . By the previous argument and the fact that $\text{SA}_t(K) \subseteq \text{Las}_t(K)$, we know that

$y_{ij} = 0$. This implies that vectors corresponding to i and j are orthogonal $v_i \cdot v_j = 0$. Further, we can express our objective function as:

$$\sum_i x_i = \sum_i y_i = \sum_i v_i \cdot v_i = \sum_i \|v_i\|^2 \tag{11}$$

Exercise 3.2. Show that $\sum_i \|v_i\|^2 \leq 1$ when there exists a unit vector v_0 such that $v_i \cdot v_0 = \|v_i\|^2$ for all $i \in [n]$ and $v_i \perp v_j$ for all $i \neq j \in [n]$.

Hence, we see that we are getting much more power by using Lasserre hierarchy as compared to Sherali Adams hierarchy. If the size of knapsack is $k - \epsilon$, then k rounds of Lasserre are needed. This is captured by the following lemma that shows that when K has “sparse support” on integral solutions, Lasserre hierarchy can discover that. Let $\text{ones}(x) = \{i \in [n] | x_i = 1\}$ be the set of indices that are one for a vector x .

Lemma 3.1. Let $K = \{x \in \mathbb{R}^n | Ax \geq b\}$ and suppose $\max_{x \in K} \{|\text{ones}(x)|\} \leq t$, then after $t + 1$ levels of Lasserre we recover the convex hull of integral solutions $\text{Las}_{t+1}^{\text{proj}}(K) = \text{conv}(K \cap \{0, 1\}^n)$

Note that the example that we did had elements with large size, if the size of the elements was small compared to the knapsack size, then the gap would have been bad. Next, we see a more general version of this theorem.

Theorem 3.1 (Decomposition Theorem [KMN11]). Let $0 \leq k \leq t$, $y \in \text{Las}_t(K)$, $S \subset [n]$ and $k \geq |\text{ones}(x) \cap S|$ then $y \in \{z | z \in \text{Las}_{t-k}(K); z_i \in \{0, 1\} \ \forall i \in S\}$

If we want to apply this to knapsack problem, we can consider S as the set of large items, i.e. with large values y_i .

3 Scheduling Problem

We will consider the scheduling problem where we are given n unit time jobs with some precedence constraints and m machines and we want to produce a schedule where the time when the last job finishes is minimized. If $i \prec j$, then i should complete before j is started. This problem was first introduced by Ron Graham.

A list-scheduling algorithm, is an algorithm that has a priority over jobs (sorted list) and always schedules the next available job in the list. Graham showed that any kind of list scheduling algorithm gives a $2 - \frac{1}{m}$ approximation algorithm for this problem. Coffman and Graham [CG72] in 1972 showed that when $m = 2$, scheduling can be done in polynomial time via matching. The approximation ratio for problem with m machines was later improved to $2 - \frac{2}{m}$. Rothvoss and Levy [LR16] showed that it is possible to get a $1 + \epsilon$ approximation algorithm for this problem which runs in $n^{\text{poly log}(n)}$ time. It is an open problem whether this can be done in polynomial time. Here, we will now recover the polynomial time algorithm for this problem with 2 machines using Lasserre hierarchy.

3.1 LP relaxation

We consider the LP relaxation of the problem. We will consider this as a feasibility problem where time goes from 1 to T and we want to schedule all the jobs within time T . Let x_{it} denote the variable whether job i was scheduled at time t .

$$\sum_{t=1}^T x_{it} = 1 \quad \forall i \quad (12)$$

$$\sum_{j \in J} x_{jt} \leq 2 \quad \forall t \quad (13)$$

$$\sum_{t \leq t'-1} x_{it} \geq \sum_{t \leq t'} x_{jt} \quad \forall i \prec j \quad (14)$$

$$x_{jt} \geq 0 \quad (15)$$

The third constraint indicates that if job j is completed by time t' , then job i should have been completed by time $t' - 1$.

Exercise 3.3. Show that if in the basic LP we used the weaker constraints:

$$\sum_{t \leq t'-1} x_{it} \geq x_{jt'}, \quad \forall i \prec j \quad (16)$$

then for any solution $y \in \text{Las}_t(K)$ for $t \geq 3$ it would be true that:

$$\sum_{t \leq t'-1} x_{it} \geq \sum_{t \leq t'} x_{jt'}, \quad \forall i \prec j \quad (17)$$

Hint: use the decomposition theorem on an appropriately chosen set S .

3.2 Integrality Gap

We show that the above LP always underestimates the time required to schedule all the jobs. Assume that $n = 6$ and the precedence constraints form a complete bipartite graph between $V_L = \{1, 2, 3\}$ and $V_R = \{4, 5, 6\}$. Each of jobs 1 – 3 should be completed before running each of jobs 4 – 6. The optimal integral solution requires $T = 4$. However, LP can schedule them in 3 time slots and thus, we have a gap of at least $4/3$. Basically, LP puts $2/3$ of jobs 1, 2, 3 in time slot 1 and then $1/3$ of 1, 2, 3 jobs and $2/3$ of 4, 5, 6 jobs in time slot 2 and rest of $2/3$ of jobs 4, 5, 6 on time slot 3. We leave it as an exercise to the reader to verify that such an assignment satisfies all the constraints.

3.3 List Scheduling and Lasserre

By using Lasserre hierarchy, we will be able to get around this problem. People in this area generally use the notion of α -completion time which denotes the time when α fraction of the job has completed. We will look at 1-completion time when the job has been fully completed.

Given a solution $y \in \text{Las}_3(K)$, we compute the completion times for all jobs j and without loss of generality we assume that the jobs are ordered according to their completion time $c_1^* \leq c_2^* \leq \dots \leq c_n^*$. To produce a schedule we then apply list-scheduling according to this ordering. This is a polynomial time algorithm and we will show that this produces a valid schedule that can be completed in $T = 3$ time slots.

Claim 3.1. *If $i \prec j$, then $c_i^* \leq c_j^* - 1$*

Proof. If we look at the constraint for job j , it says job i should be completed within time $c_j^* - 1$ if job j completes at time c_j^* . \square

Let σ_i be the time when job i is scheduled by the greedy algorithm

Claim 3.2. $\forall \text{ jobs } \sigma_j \leq c_j^*$

and for an index j , let $J := \{i | \sigma_i < \sigma_j\}$ be the set of items scheduled before j by the list scheduling algorithm.

Proof. We will do a proof by contradiction. Let us say that j_1 is the lowest indexed job where this claim is violated or $\sigma_{j_1} \geq c_{j_1}^* + 1$. Let $j_0 \in \{1, 2, \dots, j_1 - 1\}$ be the last last job scheduled without any other job in $\{1, 2, \dots, j_1\}$ scheduled in parallel. Let $K = \{j | j \leq j_1 \text{ and } \sigma_j > \sigma_{j_0}\}$ scheduled between j_0 and j_1 . Also, consider the complete time interval $[\sigma_{j_0}, \sigma_{j_1} - 1]$ of length $\sigma_{j_1} - 1 - \sigma_{j_0} = k$. By choice of j_0 , all the slots on this time interval can not be empty as otherwise j_0 would be at that slot. Also, it cannot happen that some other job after j_1 is scheduled here in this region because then also j_0 would be at that slot. The complete time interval $[\sigma_{j_0}, \sigma_{j_1} - 1]$ of length $\sigma_{j_1} - 1 - \sigma_{j_0} = k$ is thus fully busy with $2k$ jobs from K .

We next make the following observation: all the jobs in K are dependent on j_0 because if not, then that job would have been scheduled along with j_0 . Next, we use properties of Lasserre solutions.

If we condition on $x_{j_0 c_{j_0}^*} = 1$ the support of the solution can only decrease or remain same, it cannot increase. Further, we know that:

$$\sigma_{j_0} \leq c_{j_0}^* \quad \text{and} \quad \sigma_{j_1} \geq c_{j_1}^* + 1 \quad (18)$$

All of the jobs in K come after $c_{j_0}^*$ and hence after σ_{j_0} but before σ_{j_1} , as their completion time is less than or equal to $c_{j_1}^*$ which is strictly less than σ_{j_1} . Moreover, j_1 must also be scheduled before σ_{j_1} by the same reason. This is a contradiction because there are many jobs on K that cannot be fit on the time interval of length k . \square

4 Max Cut

We will talk about Max Cut and we will see how applying the Lasserre Hierarchy gives rise to known algorithms. Max cut is the problem where we are given a graph $G(V,E)$ and we want to partition the graph into two components maximizing the weight of edges cut.

4.1 LP relaxation

Now, we try to write a linear program associated with max cut. Let $x_i \in \{0, 1\}$ denote the variable for vertex i and let z_{ij} denote the variable for edge (i, j) . Hence, the way to write a linear program would be to:

$$\max \quad \sum c_{ij} z_{ij} \quad (19)$$

$$\text{s.t.} \quad z_{ij} \geq x_i - x_j \quad (20)$$

$$z_{ij} \geq x_j - x_i \quad (21)$$

$$z_{ij} \leq x_i + x_j \quad (22)$$

$$z_{ij} \leq 2 - x_i - x_j \quad (23)$$

Basically, these last 4 constraints are our way of adding the constraint $z_{ij} = |x_i - x_j|$. However, this LP is bad as we can set each $x_i = 1/2$ and each $z_{ij} = 1$ and cut all the edges fractionally.

4.2 Lassere Hierarchy

We next show how can one recover the SDP relaxation of Max-Cut used by Goemans and Williamson through Lassere Hierarchy. We consider the basic LP K and consider a solution $y \in \text{Las}_3(K)$.

We first observe that $z_{ij} = x_i + x_j - 2x_{ij}$ as every 2 vertices form a consistent probability distribution. Also, we know that matrix X with (i, j) entry as x_{ij} , we have a positive semi definite matrix. Thus, we have vectors v_i for every vertex i and $v_i \cdot v_j = x_{ij}$. How can we use these vectors $\{v_i\}_{i \in [n]}$ to produce a valid assignment $x \in \{0, 1\}^n$.

Here, we cannot directly use the Goemans-Williamson rounding as these vectors are not exactly the same. Goemans-Williamson vectors were the vector analog of $\{-1, 1\}$ variables whereas these variable are the vector analog of $\{0, 1\}$ variables. These vectors here are not even unit vectors. However, we can use the geometric properties of these vectors that we saw earlier. We can write $v_i = \frac{1}{2}(v_\phi + u_i)$ or $u_i = 2v_i - v_\phi$. These u_i are unit vectors and they correspond to the Goemans Williamson unit vectors.

$$u_i \cdot u_j = (2v_i - v_\phi) \cdot (2v_j - v_\phi) \quad (24)$$

$$u_i \cdot u_j = 4v_i \cdot v_j - 2v_i - 2v_j + 1 \quad (25)$$

$$u_i \cdot u_j = 4x_{ij} - 2x_i - 2x_j + 1 \quad (26)$$

$$u_i \cdot u_j = 1 - 2z_{ij} \quad (27)$$

$$z_{ij} = \frac{1 - u_i \cdot u_j}{2} \quad (28)$$

Thus, our objective can be re written as $\max \sum_{ij \in E} c_{ij} \left(\frac{1 - u_i \cdot u_j}{2} \right)$. If $u_i \in \mathbb{R}^d$ and $u_j \in \mathbb{R}^d$ have an angle of θ between them, then contribution of edge (i, j) can be written as $c_{ij} \left(\frac{1 - \cos(\theta)}{2} \right)$.

Now, we can separate them by random hyperplane rounding. Let $G \sim N(0, I_d)$ then for all $i \in [n]$. we set

$$x_i = \frac{1}{2}(\text{sign}(u \cdot G) + 1) \in \{0, 1\} \quad (29)$$

We can then show that:

$$\Pr(i \text{ and } j \text{ are separated}) = \frac{\theta_{ij}}{\pi} \quad (30)$$

$$\mathbb{E}[\text{cut}] = \sum c_{ij} \frac{\theta_{ij}}{\pi} \quad (31)$$

$$\text{LP contribution} = \sum c_{ij} \frac{1 - \cos(\theta_{ij})}{2} \quad (32)$$

$$\frac{\mathbb{E}[\text{cut}]}{\text{LP contribution}} = \min_{\theta} \frac{\theta/\pi}{\frac{1 - \cos(\theta)}{2}} \approx 0.878 \quad (33)$$

5 Global Correlation Rounding

Arora, Barak and Steurer [ABS10] showed that solving Unique Games or Small set expansion problem can be done in subexponential time $\approx 2^{n^{1/3}}$. This was an important result which gave an indication that this problem is likely not NP hard. Later, this result was obtained independently by Barak, Raghavendra and Steurer [BRS11] and Guruswami and Sinop [GS11] using Lasserre hierarchy.

The main idea was that if the variables in the Lasserre solution were independent, then we could just round each variable independently by setting $x_i = 1$ with probability x_i and this should give us a good solution. However, the fractional solution is very correlated for example, in max cut, variables have negative correlation with each other. The idea was to condition on variables again and again so that the correlation decreases and we get an almost independent solution. This technique is known as global correlation rounding and we will see this technique in context of max cut on dense graphs.

5.1 Max cut on dense graphs

Dense graphs have $\Omega(n^2)$ edges. There exists a PTAS [AKK95] for finding max cut on dense graphs i.e. for a fixed ϵ , we can find a solution with value $(1 - \epsilon)\text{OPT}$ in time $n^{f(\epsilon)}$. This came up in 1990's. Here, we will use the technique of *global correlation rounding* to get a PTAS for max cut on dense graphs.

Let $y \in \text{Las}_t(K)$. First, we introduce some notations and definitions:

$$\text{var}[X_i] = E[X_i^2] - E[X_i]^2 = y_i(1 - y_i) \quad (34)$$

$$\text{cov}[X_i, X_j] = E[X_i X_j] - E[X_i]E[X_j] = y_{ij} - y_i y_j \quad (35)$$

Proposition 3.1 (Law of total variance). $\mathbb{E}_{X_j}[\text{var}(X_i | X_j)] = \text{var}(X_i) - \text{var}_{X_j}[\mathbb{E}[X_i | X_j]]$.

Further, using properties of Lasserre we know that:

$$\mathbb{E}[X_i | X_j = 1] = \frac{y_{ij}}{y_j} \quad \text{and} \quad \mathbb{E}[X_i | X_j = 0] = \frac{y_i - y_{ij}}{1 - y_j} \quad (36)$$

Using this fact we can show that $\mathbb{E}_{X_j}[\mathbb{E}[X_i | X_j]] = \mathbb{E}[X_i] = y_i$ and

$$\text{var}_{X_j}[\mathbb{E}[X_i | X_j]] = y_j \frac{y_{ij}^2}{y_j} + (1 - y_j) \frac{y_i - y_{ij}}{1 - y_j}^2 - y_i^2 = \frac{y_i y_j - y_{ij}^2}{y_j(1 - y_j)} = \frac{\text{cov}(X_i, X_j)^2}{\text{var}(X_j)} \geq 4\text{cov}(X_i, X_j)^2 \quad (37)$$

This is saying that if we condition on a variable X_j , variance of the other variable decreases and the expected value of decrease is greater than $4cov(X_i, X_j)^2$. So, more the correlation among the variables, more the variance of i decreases when conditioned on j .

Let $R \subseteq [n]$, we define the variance of y w.r.t. variables in R as:

$$V_R(y) \triangleq \sum_{i \in R} y_i(1 - y_i) \quad (38)$$

We define the correlation of y w.r.t. variables in R as:

$$C_R(y) \triangleq \sum_{i \in R} \sum_{j \in R} (y_i y_j - y_{ij})$$

Theorem 3.2 (Global Correlation). *Let $K = \{X \in \mathbb{R}^n \mid Ax \geq b\}$ and $y \in Las_t(K)$ such that $t \geq \frac{1}{\epsilon^3} + 2$ and let $R \subseteq [n]$, one can condition on $\frac{1}{\epsilon^3}$ variables in R to get $y' \in Las_{t-\frac{1}{\epsilon^3}}(K)$ such that $C_R(y') \leq \frac{\epsilon^3}{4}|R|^2$. In particular, $Pr_{i,j \in R}[|y'_i y'_j - y_{ij}| \geq \epsilon] \leq \epsilon$*

This means that we can get a solution y' where only ϵ fraction of the pairs of variables have more than ϵ correlation amongst them.

Suppose, we choose a variable $j \in R$ uniformly at random and condition on that variable being a where $a = 1$ with probability y_j and 0 with probability $1 - y_j$. Let $y' \in Las_{t-1}(K)$ be the solution obtained by this conditioning, then $\mathbb{E}_{j,a}[V_R(y')] = V_R(y) - \frac{4C_R(y)}{|R|}$. If initial correlation is high, variance keeps going down. We run the following procedure:

1. Repeat
2. If $C_R(y') \leq \frac{\epsilon^3}{4}|R|^2$ return y'
3. Find $j \in R$ and $a \in \{0, 1\}$ such that conditioning reduces variance by the largest amount.

The reader should think about that whether we need to select the variable which decreases the variance by the largest amount or we can select variables uniformly randomly. When we do this conditioning, variance is going down but the objective value is not going down. Now, after repeating this procedure, we can round each of the variable independently by setting $x_i = 1$ with probability y_i and 0 otherwise. The Lasserre objective gets y_{ij} and we get $y_i y_j$. But, this is the same for all but ϵ fraction of the pairs by the global correlation theorem. Thus, we obtain a good solution for max cut on dense graphs.

Note: However, this will not always work for sparse graphs as we do not know which ϵ fraction of the pairs will go bad and the graph might have put all the weight on just those edges.

References

- [ABS10] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, FOCS '10*, pages 563–572, Washington, DC, USA, 2010. IEEE Computer Society.

- [AKK95] Sanjeev Arora, David Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of np-hard problems. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 284–293. ACM, 1995.
- [BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 472–481. IEEE, 2011.
- [CG72] Edward G Coffman and Ronald L Graham. Optimal scheduling for two-processor systems. *Acta informatica*, 1(3):200–213, 1972.
- [GS11] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with psd objectives. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 482–491. IEEE, 2011.
- [KMN11] Anna R Karlin, Claire Mathieu, and C Thach Nguyen. Integrality gaps of linear and semi-definite programming relaxations for knapsack. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 301–314. Springer, 2011.
- [LR16] Elaine Levey and Thomas Rothvoss. A $(1+\epsilon)$ -approximation for makespan scheduling with precedence constraints using lp hierarchies. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing, STOC '16*, pages 168–177, New York, NY, USA, 2016. ACM.
- [Rot13] Thomas Rothvoß. The lasserre hierarchy in approximation algorithms. *Lecture Notes for the MAPSP*, pages 1–25, 2013.