Generalized Lagrangian Duals and Sums of Squares Relaxations of Sparse Polynomial Optimization Problems

Hayato Waki (T.I.T, Japan)
Sunyoung Kim (Ewha Women's Univ, Korea)
Masakazu Kojima (T.I.T, Japan)

Contents

- Generalized Lagrangian Dual
- Numerical experiments
- Heuristics for POP with special structure

Generalized Lagrangian Duals

Our purpose I

Polynomial Optimization Problem

POP $\left\{ \begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_j(x) \geq 0 \quad (j = 1, \ldots, m) \end{array} \right.$

Lagrange Dual $\displaystyle \max_{\psi \in \mathbb{R}^n_+} \min_{x \in \mathbb{R}^n} \left\{ f(x) - \sum_{j=1}^{m} \psi_j g_j(x) \right\}$

▷ Size of SDP by Lasserre's SDP relax is very large.

▷ Apply Lasserre's SDP relaxation to Lagrangian Dual

⇒ Size of SDP is smaller.

⇒ Opt.Val.of Lagrangian Dual is obtained.

Our purpose II

▷ **But, in general,**

Opt.Val. of Lagrangian Dual $\leq$ Opt.Val. of given POP.

$\Rightarrow$ *Can not expect to get Opt.Val. of POP by this approach*

▷ **Improve Lagrangian func. in order to get better value.**

$\Rightarrow$ **Generalized Lagrangian function**

Polynomial Optimization Problem

$$\textbf{(POP)} \quad \begin{cases} \min & f(x) \\ \text{s.t.} & g_j(x) \geq 0 \quad (j = 1, \ldots, m) \end{cases}$$

Generalized Lagrangian function $L(x, \psi)$

$$L(x, \psi) = f(x) - \sum_{j=1}^{m} g_j(x) \psi_j(x),$$

where

$$\forall x \in B_a = \{x \in \mathbb{R}^n \mid x_i^2 \leq a \quad (\forall i)\}$$

$$\forall \psi_j \in \textbf{SOS} := \left\{ p \in \textbf{Poly.} \,\middle|\, p = \sum_{j=1}^{K} (\textbf{Poly.})^2 \right\} \quad (\textbf{Sums Of Squares})$$

$$\psi = (\psi_1, \ldots, \psi_m)^T$$

$L(x, \psi)$

$$L(x, \psi) = f(x) - \sum_{j=1}^{m} g_j(x)\psi_j(x) \quad (\forall x \in B_a, \forall \psi_j \in \textbf{SOS})$$

Property of $L(x, \psi)$

- **Extension of Lagrangian function since $\mathbb{R}_+ \subset$ SOS**
- **Opt.Val. of POP $\geq \max_{\psi \in \textbf{SOS}} \min_{x} L(x, \psi)$: (Generalized Lagrangian Dual)**

Result

- Regard $-\sum_{j=1}^{m} g_j(x)\psi_j(x)$ as Penalty func,...

$$\max_{\psi \in \mathbf{SOS}} \min_{x} L(x, \psi) = (\textbf{Opt.Val. of POP})$$

- Apply SDP relaxation for unconstrained POP,...

  $\Rightarrow$ same as the SDP relaxation for constrained POP

- Using the sparsity of poly., reduce the size of SDP.

$P(x,\psi)$: Second term of $L(x,\psi)$

$$L(x,\psi) = f(x) - \sum_{j=1}^{m} g_j(x)\psi_j(x) \quad (\forall x \in B_a, \forall \psi_j \in \textbf{SOS})$$

$$P(x,\psi) = -\sum_{j=1}^{m} g_j(x)\psi_j(x) \quad (\forall x \in B_a, \forall \psi_j \in \textbf{SOS})$$

Property of $P(x,\psi)$

- $\exists \psi^k = \left(\psi_1^k, \ldots, \psi_m^k\right)^T \in \textbf{SOS}^m \quad (k = 1, 2, \ldots,)$

$\Rightarrow$ s.t. $P(x,\psi^k) \to \mathbf{I}(x) = \begin{cases} 0 & (x \in \text{Feasible region}) \\ +\infty & (\text{otherwise}) \end{cases} \qquad (k \to \infty)$

We can regard $P(x,\psi)$ as Penalty func.

k sufficient large...

- $L(x, \psi^k) = f(x) + P(x, \psi^k) \approx \begin{cases} f(x) & (x \in \text{Feasible region}) \\ \text{large value} & (\text{otherwise}) \end{cases}$

$\Rightarrow \min_{x} \left\{ f(x) + P(x, \psi^k) \right\}$ has $\begin{cases} \text{Opt.Sol. } x_k \in \text{Feasible Region} \\ \text{Opt.Val } f(x_k) \end{cases}$

$\Rightarrow \lim_{k \to \infty} \min_{x} \left\{ f(x) + P(x, \psi^k) \right\} = \text{Opt.Val.}$

$\Rightarrow \max_{\psi \in \text{SOS}} \min_{x} \{ f(x) + P(x, \psi) \} = (\text{Opt.Val.})$

Apply SDP relaxation

1. fix $\bar{\psi} \in$ **SOS**.
2. Apply SDP relaxation for unconstrained POP.

$$\left\{\begin{array}{ll}\min & L(x,\bar{\psi}) \\ \text{s.t.} & x \in \mathbb{R}^n\end{array}\right. \equiv \left\{\begin{array}{ll}\max & \eta \\ \text{s.t.} & L(x,\bar{\psi}) - \eta \geq 0 \quad (\forall x \in B_a)\end{array}\right.$$

$$\Rightarrow \left\{\begin{array}{ll}\max & \eta \\ \text{s.t.} & L(x,\bar{\psi}) - \eta \in \text{SOS} + \sum_{i=1}^n (a - x_i^2)\text{SOS}\end{array}\right.$$

3. Regard $\bar{\psi}$ as variable.

$$\equiv \left\{\begin{array}{ll}\max & \eta \\ \text{s.t.} & f(x) - \sum_{j=1}^m g_j(x)\psi_j(x) - \eta = \psi_0(x) + \sum_{i=1}^n (a - x_i^2)\phi_i(x) \quad (\\ & \psi_0, \psi_j \in \text{SOS}, \phi_i \in \text{SOS}.\end{array}\right.$$

"Penalty Function" + "SDP relax for unconstrained POP"

= "SDP relax for constrained POP".

<u>But, the size of SDP is much large...</u>

Reduce the size of SDP

- Let

$$
\begin{aligned}
C_j &= \left\{ i \in \{1, \ldots, n\} \,\middle|\, x_i \text{ appears in } g_j \right\} \\
\text{SOS}_j &= \left\{ p \in \text{SOS} \,\middle|\, p = \sum_{k=1}^{\ell} q_k(x, C_j)^2 \right\},
\end{aligned}
$$

where $q_k(x, C_j)$ is represented by the variable $x_i \in C_j$.

- For example,

$$
\begin{aligned}
g_2(x) &= -2 - x_2 x_3 + {x_3}^2 \Rightarrow C_j = \{2, 3\} \\
&\Rightarrow \text{SOS}_2 := \left\{ p \in \text{SOS} \,\middle|\, p(x) = \sum_{i=1}^{\ell} q_i(x_2, x_3)^2 \right\}
\end{aligned}
$$

Reduce the size of SDP II

some function ψ_j^k

$$\psi_j^k(x) = \left(1 - \frac{g_j(x)}{\gamma}\right)^{2k} \Rightarrow P(x, \psi^k) \rightarrow \textbf{Indicator}(x)$$

- $\{i \mid x_i$ **appears in** $\psi_j^k\} = C_j$

$$\Rightarrow \psi_j^k \in \textbf{SOS}_j = \left\{ p \in \textbf{SOS} \,\middle|\, p = \sum_{k=1}^{\ell} q_k(x, C_j)^2 \right\} \quad (\forall k)$$

- $\max_{\psi_j \in \textbf{SOS}_j} \min_{x \in \mathbb{R}^n} L(x, \psi) =$ **Opt.Val.**

- g_j **has few variables** $\Rightarrow$ $\textbf{SOS}_j$ **is much smaller set**$\Rightarrow$**The size of SDP by** $\textbf{SOS}_j$ **is smaller.**

Reduce the size of SDP III

- But, theoretically, convergence is *not* guaranteed...

$$\lim_{k\to\infty} \text{(Opt.Val. of SDP)} \stackrel{?}{=} \text{(Opt.Val of POP)}$$

- Need to reduce the following set.

$$L(x,\psi) - \eta = f(x) - \sum_{j=1}^{m} \psi_j(x) g_j(x) - \eta \in \textbf{SOS}.$$

- Given POP has specail structure ⇒ Can generate much smaller SDP (from this SOS).

Numerical experiments

Example 1

$$\begin{cases} \min & \sum_{i=1}^{n-1} \sum_{j+\ell\leq 4, j,\ell\geq 0} a_{ij\ell} x_i^j x_{i+1}^\ell \\ \text{s.t.} & (x_i, x_{i+1})^T Q_{ij} \begin{pmatrix} x_i \\ x_{i+1} \end{pmatrix} \leq 1 \quad (i = 1,\ldots,n-1, j = 1,\ldots,n) \\ & (Q_{ij} \succeq O\text{: constant matrix}) \end{cases}$$

$$\psi_i \in \mathbf{SOS}_i = \left\{ p \in \mathbf{SOS} \,\middle|\, p(x) = \sum_{k=1}^{K} q_k(x_i, x_{i+1})^2 \right\}$$

Enviroment of Computer...

CPU is Pentium IV(Xeon) and memory 6GB. Solver is SDPA.

Result of Example 1

- Both relaxation attain Opt.Val. at $k = 2$

- Our relaxation is much faster than Lasserre's.

	cpu time[sec]	
n	Lasserre	Ours
10	18.44	9.50
11	45.59	14.88
12	112.48	32.86
13	895.87	166.07
14	1527.50	395.47

Example 2

$$\begin{cases} \min \ \sum_{i=1}^{n-1} \sum_{j+\ell \le 6, j,\ell \ge 0} a_{ij\ell} x_1^j x_{i+1}^\ell \\ \text{s.t.} \quad \begin{array}{llllll} x_1^6 & +x_2^2 & & & \le & 1, \\ x_1^6 & & +x_3^2 & & \le & 1, \\ \vdots & & & \ddots & & \vdots, \\ x_1^6 & & & +x_n^2 & \le & 1. \end{array} \end{cases}$$

$$\psi_i \in \mathbf{SOS}_i = \left\{ p \in \mathbf{SOS} \,\middle|\, p(x) = \sum_{\ell=1}^{K} q_\ell(x_1, x_{i+1})^2 \right\}$$

Result of Example 2

- Both relaxation attain Opt.Val. at $k = 5$

- small difference of cpu time between Lasserre's and ours

	cpu time[sec]	
n	Lasserre	Ours
3	3.12	2.83
4	61.91	39.12
5	1542.28	1468.68

Why small Difference?

SDP relaxed problem

$$\begin{cases} \max & \eta \\ \text{s.t.} & f(x) - \eta = \psi_0(x) + \sum_{j=1}^{m} g_j(x)\psi_j(x) \\ & \psi_0 \in \mathbf{SOS}, \psi_j \in \mathbf{SOS}_j. \end{cases}$$

- **The size of variable matrix induced from SOS is the largest.**

⇒ **reduce the size by using** *the structure...*

Example 2

$$\begin{cases} \max & \eta \\ \textbf{s.t.} & f(x) - \sum_{j=1}^{n-1} \psi_j(x_1, x_{j+1}) g_j(x_1, x_{j+1}) - \eta = \psi_0(x) \quad (\forall x \in \mathbb{R}^n) \\ & \psi_0 \in \textbf{SOS}, \psi_j \in \textbf{SOS}_j. \end{cases}$$

- $f(x) = \sum_{j=1}^{n-1} a_{jk\ell} x_1^k x_{j+1}^\ell$
- **Only $x_1^k x_{j+1}^\ell$ appears in the left-hand side of identity.**
- **By observation, $\exists ? \psi_0$ satisfying identity even if ψ_0 has only $x_1^k x_{j+1}^\ell$**
- **Replacing $\psi_0(x)$ by $\sum_{j=1}^{n-1} \psi_{0j}(x_1, x_{j+1}) \quad \psi_{0j} \in \textbf{SOS}_j$**

Result of Lasserre, Ours and Heuristics

- Both relaxation attain Opt.Val. at $k = 5$
- Difference between Lasserre and heuristics $\nearrow$, as $n \nearrow$.

	cpu time[sec]		
n	Lasserre	Ours	Heuristics
3	3.12	2.83	0.15
4	61.91	39.12	0.46
5	1542.28	1468.68	1.46
50	—	—	160.62

Example 3

$$\begin{cases} \min \sum_{i=1}^{n-1} \sum_{j+\ell \leq 6, j,\ell \geq 0} a_{ij\ell} x_i^j x_{i+1}^\ell \\ \text{s.t.} \quad \begin{array}{llllll} x_1^6 & +x_2^2 & & & \leq & 1, \\ & x_2^6 & +x_3^2 & & \leq & 1, \\ & & \ddots & \ddots & \leq & 1, \\ & & & x_{n-1}^6 \quad +x_n^2 & \leq & 1. \end{array} \end{cases}$$

At $k = 5$, all relaxations attain optimal value.

		cpu time[sec] (optimal value)	
n (optimal value)	k	Lasserre	Heuristics
3 (-2.437)	4	0.49 (-2.825)	0.14 (-2.860)
	5	1.43 (-2.437)	0.16 (-2.437)
4 (-3.711)	4	4.71 (-3.711)	0.13 (-3.711)
5 (-5.191)	4	82.29 (-5.578)	0.34 (-5.604)
	5	908.78 (-5.191)	0.44 (-5.191)

Conclusion

- “Penalty Function” + “SDP relax for unconstrained POP”
  = “SDP relax for constrained POP”.

- Using the structure of problem, Heuristics is very efficient

- Opt.Val *may* be abtained by Heuristics.