Non-convex quadratically constrained quadratic programming

Jing Guo
Department of Computer Science
Shanghaitech University
2019533173
guojing@shanghaitech.edu.cn

Zixuan Shu
Department of Computer Science
Shanghaitech University
2019533150
shuzx@shanghaitech.edu.cn

Ziyun Zeng
Department of Computer Science
Shanghaitech University
2019533086
zengzy1@shanghaitech.edu.cn

Abstract

The future IoT will have billions of sensors. A technique called over-the-air function are used in order to overcome the challenge that the traditional "aggerate-then-compute" is too slow. In this algorithm, we simplize the function to a non-convex QCQP. We apply three methods to solve the optimal problem. Which are "SDR and SCA", "DCA" and "FPP-SCA".

1 Introduction

In the future, the Internet-of-Things(IoT) will connect billions of sensors. A technique called over-the-air function is used since the traditional method "aggerate-then-compute" is too slow. This method use the superposition property of wireless channel. As we can see in the figure 1. The fusion center(FC)

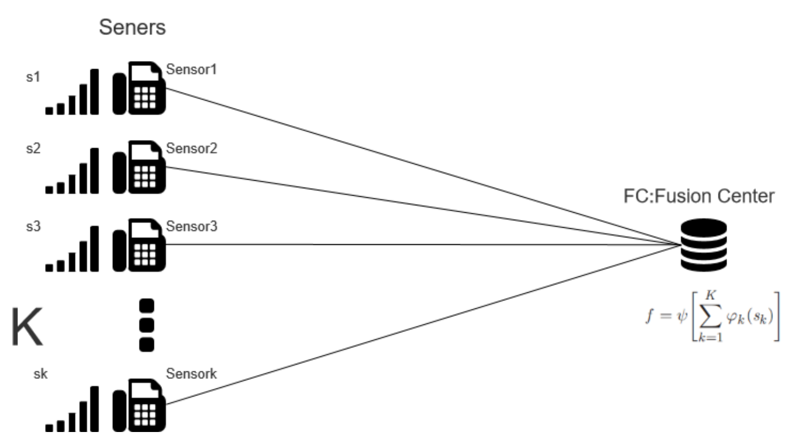

Figure 1: The future IoT

receives the signals send by the sensors. [1]

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

1.1 Model Defination

The target function canbe written as

$$f = \psi\left[\sum_{k=1}^{K}\varphi_k\left(s_k\right)\right] \tag{1}$$

Where the φ_k is the pre-processing function while the ϕ_k is the post-processing function.

First we let

$$x_k = \varphi_k\left(s_k\right). \tag{2}$$

And our goal is to estimate

$$g = \sum_{k=1}^{K}\varphi_k\left(s_k\right) = \sum_{k=1}^{K}x_k, \tag{3}$$

But we receive the signal after transmissions of all the sensors and it is

$$\mathbf{y} = \sum_{k=1}^{K}\mathbf{h}_k b_k x_k + \mathbf{z} \tag{4}$$

Where $\mathbf{h}_k$ is the channel vector, b_k is the transmitter scalar and $\mathbf{z}$ is the noise.

Then the estimated function seems like

$$\hat{g} = \mathbf{a}^T\mathbf{y} = \mathbf{a}^T\sum_{k=1}^{K}\mathbf{h}_k b_k x_k + \mathbf{a}^T\mathbf{z}, \tag{5}$$

Where a is the receive vector.

At last, we compute the efficiency using mean squared error(MSE),

$$\mathrm{MSE}(\hat{g}, g) = \mathrm{E}\left(|\hat{g} - g|^2\right). \tag{6}$$

[1]

1.2 Transmitter Design

Now we design the transmitter to get b_k.In order to estimate with less variables, we degine a uniform-forcing transmitter like

$$b_k = \sqrt{\eta}\frac{\left(\mathbf{a}^T\mathbf{h}_k\right)^H}{\left\|\mathbf{a}^T\mathbf{h}_k\right\|^2}, \tag{7}$$

Where

$$\eta = P_0\min_k\left\|\mathbf{a}^H\mathbf{h}_k\right\|^2, \tag{8}$$

Using b_k to substitute the one in equation(4) and we can get

$$\hat{g} = \sum_{k=1}^{K}x_k + \frac{\mathbf{a}^H\mathbf{z}}{\sqrt{\eta}}. \tag{9}$$

Thus the MSE can be simplized to an optimal problem

$$(\mathrm{P1.1})\ \ \min_{\mathbf{a}}\max_k\ \frac{\|\mathbf{a}\|^2}{\left\|\mathbf{a}^H\mathbf{h}_k\right\|^2}. \tag{10}$$

We substitute $\|\mathbf{a}\|$ with $\frac{\|\mathbf{a}\|}{\sqrt{\tau}}$ where $\tau = \min_k\left\|\mathbf{a}^H\mathbf{h}_k\right\|^2$ and we have P1.2, which is a non-convex quadratic constraint quadratic problem(QCQP).

$$\begin{aligned}(\mathrm{P1.2})\ \ &\min_{\mathbf{a}}\ \|\mathbf{a}\|^2\\ &\text{s.t. } \left\|\mathbf{a}^H\mathbf{h}_k\right\|^2 \geq 1,\ \forall k.\end{aligned} \tag{11}$$

Here we give three method to deal with P1.2. [1]

2 Algorithms

2.1 SDR and SCA

First, we apply the semidefinate relaxation(SDR) to get a. Since P1.2 is non-convex, we relax it to a convex SDP first.

We let $A = aa^H$ and $H_k = h_k h_k^H$

$$\begin{aligned}(\text{P2.1}) \min_{\mathbf{A}} \ & \text{tr}\left(\mathbf{A}\right) \\ \text{s.t.} \ & \text{tr}\left(\mathbf{A}\mathbf{H}_k\right) \geq 1, \ \forall k \\ & \mathbf{A} \succeq \mathbf{0}, \text{rank}\left(\mathbf{A}\right) = 1. \end{aligned} \tag{12}$$

However, this problem is non-convex since rank(A)=1 is a non-convex constraint. In this case, we just delete this constraint and get P2.2.

$$\begin{aligned}(\text{P2.2}) \min_{\mathbf{A}} \ & \text{tr}\left(\mathbf{A}\right) \\ \text{s.t.} \ & \text{tr}\left(\mathbf{A}\mathbf{H}_k\right) \geq 1, \ \forall k, \ \mathbf{A} \succeq \mathbf{0}. \end{aligned} \tag{13}$$

Which is convex and an optimal solution can be found. In order to get a from A, we apply an approximate solution

$$a^* = \sqrt{\lambda_1} u_1 \tag{14}$$

where λ_1 is the maximum eigen value of A while u_1 is the corresponding eigen vector.

Since the solution A may not be rank=1, the performance of this approximate solution is not good enough. In this case, we apply SCA. We let $\mathbf{c}_k = \left[\text{Re}\left(\mathbf{a}^H \mathbf{h}_k\right), \text{Im}\left(\mathbf{a}^H \mathbf{h}_k\right)\right]$ and rewrite P1.2 to P3.1

$$\begin{aligned}(\text{P3.1}) \min_{\mathbf{c}_k} \ & \left\|\mathbf{a}\right\|^2 \\ \text{s.t.} \ & \left\|\mathbf{c}_k\right\|^2 \geq 1, \ \forall k \\ & \mathbf{c}_k = \left[\text{Re}\left(\mathbf{a}^H \mathbf{h}_k\right), \text{Im}\left(\mathbf{a}^H \mathbf{h}_k\right)\right], \ \forall k \end{aligned} \tag{15}$$

which is non-convex because $\left\|\mathbf{c}_k\right\|^2 \geq 1$ is a non-convex constraint. so we relax it to a linear constraint using 1st Talor function

$$\left\|\mathbf{c}_k\right\|^2 \geq \left\|\mathbf{c}_k^{(l)}\right\|^2 + 2\left(\mathbf{c}_k^{(l)}\right)^T \left(\mathbf{c}_k - \mathbf{c}_k^{(l)}\right) \geq 1, \forall k \tag{16}$$

and we get a convex problem P3.2

$$\begin{aligned}(\text{P3.2}) \min_{\mathbf{a}, \{\mathbf{c}_k\}} \ & \left\|\mathbf{a}\right\|^2 \\ \text{s.t.} \ & \left\|\mathbf{c}_k^{(l)}\right\|^2 + 2\left(\mathbf{c}_k^{(l)}\right)^T \left(\mathbf{c}_k - \mathbf{c}_k^{(l)}\right) \geq 1, \forall k \\ & \mathbf{c}_k = \left[\text{Re}\left(\mathbf{a}^H \mathbf{h}_k\right), \text{Im}\left(\mathbf{a}^H \mathbf{h}_k\right)\right], \forall k \end{aligned} \tag{17}$$

where l means the lth iteration of convex approximation.

In this algorithm, we use SCA to get a initial solution of the problem and use SDR to convex approximate it until the adjacent two iterations get two close solutions which means the difference between them is less then a threshold ϵ.

Here is the algorithm[1]

Algorithm 1 SDR and SCA

Input: ϵ,h_k,a
Output: the approximated optimal solution a^*

Solve P2.2 and get A^*
if rank$A^* \neq 1$ **then**
  initial solution $a^* = \sqrt{\lambda_1} u_1$
  $\mathbf{c}_k = \left[\mathrm{Re}\left(\mathbf{a}^H \mathbf{h}_k\right), \mathrm{Im}\left(\mathbf{a}^H \mathbf{h}_k\right)\right], \forall k$
  while $\sum_{k=1}^{K} \left\| c_k^{(l+1)} - c_k^{(l)} \right\| \leq \epsilon$ **do**
    Solve P3.2 to get $\{c_k^{(l+1)}\}, \forall k$ and a^*
else
  $a^* = \sqrt{\lambda_1} u_1$
return a^*

2.2 DCA

A. DC Formulation

The constraint $rank(A) = 1$ in P2.1 can be reformulated as $Tr(A) - \|A\|_2 = 0$, where $\|A\|_2$ denotes the spectral norm of matrix A. Since $Tr(A) = \sum_i \lambda_i$ and $\|A\|_2 = \lambda_{max}$, so $Tr(A) - \|A\|_2 >= 0$ always holds. Then we can reformed P2.1 as a difference-convex problem (DCP)[4]:

$$\begin{aligned} \text{(P4.1)} \min_{\mathbf{A}} \ & \mathrm{tr}\left(\mathbf{A}\right) + \rho\left(\mathrm{tr}\left(\mathbf{A}\right) - \|A\|_2\right) \\ \text{s.t. } & \mathrm{tr}\left(\mathbf{A}\mathbf{H}_k\right) \geq 1, \ \forall k \\ & \mathbf{A} \succeq \mathbf{0}. \end{aligned} \tag{18}$$

where $\rho > 0$ denotes the penalty parameter.

B. DC Algorithm
The formulation in P4.1 can be rewritten as $h_1 - g_1$, where $h_1 = (1+\rho)Tr(A)$, $g_1 = \|A\|_2$. By linearizing the concave part $-g_1$ in objective function, the resulting subproblem is given by[5]

$$\begin{aligned} \text{(P4.2)} \min_{\mathbf{A}} \ & (1+\rho)\mathrm{tr}\left(\mathbf{A}\right) + \rho\left\langle \partial_{A_{[t-1]}} g_1, A \right\rangle \\ \text{s.t. } & \mathrm{tr}\left(\mathbf{A}\mathbf{H}_k\right) \geq 1, \ \forall k \\ & \mathbf{A} \succeq \mathbf{0}. \end{aligned} \tag{19}$$

where $\langle X, Y \rangle$ denotes the inner product of X and Y i.e. $\langle X, Y \rangle = \left[\mathbf{tr}(X^H Y)\right]$. And $\partial_{M_{[t-1]}} g_1$ is the subgradient of g_1 for t^{th} iteration.

Algorithm 2 DC Algorithm (DCA)

Input: A, H, ρ, a small value $\epsilon > 0$
Output: the approximated optimal solution A

$t = 1$.
while $\left|f^{t-1} - f^t\right| > \epsilon$ **do**
  Select a $S_M^{t-1} \in \partial_{A_{[t-1]}} g_1$.
  Solve the convex subproblem P4.2 and obtain A$_{[t]}$. $Increment$t.

[2] The complexity of algorithm above can be computed as below:

$$\beta = KN^2 \tag{20}$$

$$C_{form} = KN^4 \tag{21}$$

$$C = \sqrt{\beta} \cdot C_{form} = K^{\frac{3}{2}} N^5 \tag{22}$$

2.3 FPP-SCA Approach

A standard QCQP can be expressed as

$$\begin{aligned} (\text{P1.1}) \min_{\mathbf{x} \in \mathbf{c}^n} & \ \mathbf{x^H} \mathbf{A}_0 \mathbf{x} \\ \text{s.t.} & \ \mathbf{x^H} \mathbf{A}_m \mathbf{x} \le c_m \\ & \ m = 1, ..., M \end{aligned} \tag{23}$$

The problem needed to solve is

$$\begin{aligned} \min_{a} \quad & \|a\|^2 \\ \text{s.t.} \quad & \|a^H h_k\|^2 \ge 1, \forall k \end{aligned}$$

It can be changed into a standard QCQP form.

$$\begin{aligned} \min_{a} \quad & a^H I a \\ \text{s.t.} \quad & a^H (-h_k h_k^H) a \le -1, \forall k \end{aligned}$$

According to SDR method, we can change the problem into

$$\begin{aligned} (\text{P1.2}) \min_{\mathbf{X} \in \mathbf{c}^{n \times n}} & \ Trace(\mathbf{A}_0 \mathbf{X}) \\ \text{s.t.} & \ Trace(\mathbf{A}_m \mathbf{X}) \le c_m, \qquad m = 1, ..., M \\ & \ X \succeq 0 \end{aligned} \tag{24}$$

Problem (P1.2) may or may not be feasible, and establishing (in)feasibility is generally NP-hard. When infeasible, one may instead seek a compromise that minimizes constraint violations in some sense - this is common in engineering applications. In order to account for potential infeasibility, consider adding slack variables s$\in \mathbf{R^M}$

$$\begin{aligned} (\text{P2}) \min_{\mathbf{x} \in \mathbf{c}^n, s \in \mathbf{R^M}} & \ \mathbf{x^H} \mathbf{A}_0 \mathbf{x} + \lambda \|s\| \\ \text{s.t.} & \ \mathbf{x^H} \mathbf{A}_m \mathbf{x} \le c_m + s_m, \\ & \ s_m \ge 0, \qquad m = 1, ..., M \end{aligned} \tag{25}$$

where λ trades off the original objective function and the slack penalty term, and $\|\cdot\|$ can be any vector norm. Problem (P2) is always feasible, and if(x_0, s_0) is an optimal solution of Problem (P2) and it so happens that s_0=0 when x_0 is an optimal solution of Problem (P1.1) Successive convex approximation (SCA) can be a method to change Problem (P2) into a convex problem. Using eigen-decomposition, the matrix $\mathbf{A}_m$ can be expressed as $\mathbf{A}_m = \mathbf{A}_m^{(-)} + \mathbf{A}_m^{(+)}$, where $\mathbf{A}_m^{(+)} \succeq 0$ and $\mathbf{A}_m^{(-)} \preceq 0$. Due to the definition of semi-definite matrix, for any x,z $\in \mathbf{C}^{n \times 1}$, $(x-z)^{\mathbf{H}} \mathbf{A}_m^{(-)} (x-z) \le 0$, then expand it.

$$\mathbf{x^H} \mathbf{A}_m^{(-)} \mathbf{x} \le 2Re\{\mathbf{z^H} \mathbf{A}_m^{(-)} \mathbf{z}\} - \mathbf{z^H} \mathbf{A}_m^{(-)} \mathbf{z} \tag{26}$$

Therefore, using the linear restriction above around the point z, we may replace the m-th (non-convex) constraint of Problem (P2) with the convex constraint

$$\mathbf{x^H} \mathbf{A}_m^{(+)} \mathbf{x} + 2Re\{\mathbf{z^H} \mathbf{A}_m^{(-)} \mathbf{z}\} \le c_m + \mathbf{z^H} \mathbf{A}_m^{(-)} \mathbf{z} + s_m \tag{27}$$

So that the problem can be reformulated into a convex problem.

$$
\begin{aligned}
(\text{P3}) \min_{\mathbf{x},s} \ & \mathbf{x^H} \mathbf{A}_0 \mathbf{x} + \lambda \sum_{m=1}^{M} s_m \\
\text{s.t. } & \mathbf{x^H} \mathbf{A}_m^{(+)} \mathbf{x} + 2Re\{\mathbf{z^H} \mathbf{A}_m^{(-)} \mathbf{z}\} \leq c_m + \mathbf{z^H} \mathbf{A}_m^{(-)} \mathbf{z} + s_m \\
& s_m \geq 0, \qquad m = 1, ..., M
\end{aligned} \tag{28}
$$

This leads us to propose the following algorithm.[3]

Algorithm 3 Feasible Point Pursuit Successive Convex Approximation (FPP-SCA) Algorithm

Initialization: Set k=0 and randomly generate an initial point z_0.

Repear:

Solve P3

Let x_k denote the optimal x obtained by solving Problem (P3) at the k-th iteration, and set $z_{k+1} = x_k$.

Set k=k+1.

until convergence.

Then compute the difficulty of FPP-SCA Algorithm.

$$
\begin{aligned}
& c \cdot \sqrt{\sum_{j=1}^{p} k_j + 2m} \cdot (\sum_{j=1}^{p} k_j^3 + c \sum_{j=1}^{p} k_j^2 + \sum_{j=1}^{m} k_j^2 + c^2) \\
& = 2 \cdot \sqrt{M(1+n)} \cdot (Mn^3 + M + 2Mn^2 + 2M + 4) \\
& \qquad = O(M^{1.5} n^{3.5})
\end{aligned} \tag{29}
$$

3 Numerical Result

We test semidefinite relaxation(SDR for shor) and successive convex approximation(SCA for short) adopted different solvers in cvx. For solvers we choose SDPT3 and MOSEK.

We measured the result by equation6 which can be written as

$$
MSE = \frac{\|a^H\|^2 \delta_n^2}{P_0 \min_k \|a^H h_k\|} \tag{30}
$$

The transmit signal to noise ratio $\frac{\delta_n^2}{P_0}$ is fixed at 20dB. And we set the threshold ϵ in algorithm1 as 1e-5.

(a) SDPT3 (b) MOSEK

Figure 2: The MSE versus different numbers of receive antennas.

The result of MSE(dB) versus different numbers of receive antennas is shown in Fig2, we fixed amount of sensors $k = 5$ and N increasing from 1 to 10 with step 1. MOSEK performed better in almost all cases, especially when $N = 1$, which means MOSEK can find a better solution only with SDR, therefore it can find the final solution more efficiently in SCA.

(a) SDPT3

(b) MOSEK

Figure 3: The MSE versus different numbers of sensors.

The result of MSE(dB) versus different numbers of sensors is shown in Fig3, we fixed amount of receive antennas $N = 4$ and sensors' number k increasing from 10 to 100 with step 10. The result computed by SDPT3 is similar with that with MOSEK. We can only tell the efficiency of SCA as shown in the paper.[1]

References

[1] Li Chen, Xiaowei Qin, and Guo Wei. "A Uniform-Forcing Transceiver Design for Over-the-Air Function Computation". In: *IEEE Wireless Communications Letters* 7.6 (2018), pp. 942–945. DOI: 10.1109/LWC.2018.2840157.

[2] J. Y. Gotoh, A. Takeda, and K. Tono. "DC formulations and algorithms for sparse optimization problems". In: *Mathematical Programming* 169.1 (2018), pp. 141–176.

[3] O Mehanna et al. "Feasible Point Pursuit and Successive Approximation of Non-Convex QCQPs". In: *IEEE Signal Processing Letters* 22.7 (2014), pp. 804–808.

[4] Hoang Tuy. *Convex Analysis and Global Optimization*. Convex Analysis and Global Optimization, 1998.

[5] Zhibin Wang et al. *Federated Learning via Intelligent Reflecting Surface*. 2020. arXiv: 2011.05051 [cs.IT].