Algebra for Machine Learning and Stochastic Programming II

Yuchen Ge

August 2022

Contents

Abstract

This article introduces cost-space scenario clustering (CSSC) method in Stochastic Optimization and how Gröbner and Graver Basis is helpful to reduce the computational difficulty in the method.

1 Problem Statement

We consider the following stochastic optimization problem:

$$\min_{x\in X\subseteq\mathbb{R}^n}\left\{f(x):=\frac{1}{N}\sum_{i=1}^{N}F\left(x,\xi_i\right)\right\},\tag{1}$$

where $\xi_1,\ldots,\xi_N$ are equiprobable scenarios taking values in $\mathbb{R}^d$ and $F\left(x,\xi_i\right)$ represents the cost associated to the decisions $x\in X$ in scenario ξ_i. The optimal solution of this problem is denoted by x^* and its optimal value by v^*. The problem is formulated using equiprobable scenarios for simplicity; the method introduced can be easily generalized to scenarios with different probabilities.

The cost function F may be given explicitly (if the problem is one-stage), or may be itself the result of a second-stage optimization problem. So

1. **In the first case, if the problem is one-stage, it can be viewed as a second-stage problem.**
2. In this latter case, which is the framework of two-stage stochastic programming, it typically takes the following form:

$$F\left(x,\xi_i\right)=\min_{y\in Y\left(x,\xi_i\right)}g\left(x,y,\xi_i\right),$$

1.1 Method of cost-space scenario clustering[1]

Suppose that problem (1) cannot be solved as it is, so we need to build from it an approximate problem composed of K scenarios with $K\ll N$. There are two broad ways to generate those scenarios:

1. they may be picked directly in the original set $\{\xi_1,\ldots,\xi_N\}$
2. they may be completely new scenarios that do not exist in the original set.

Consider for now that this set has been computed, and let us denoted it by $\left\{\widetilde{\xi}_1,\ldots,\widetilde{\xi}_K\right\}$ with the corresponding probabilities $\{p_1,\ldots,p_K\}$. The approximate problem takes the form:

$$\min_{x\in X\subseteq\mathbb{R}^n}\left\{\widetilde{f}(x):=\sum_{k=1}^{K}p_kF(x,\widetilde{\xi}_k)\right\}$$

and its optimal solution is denoted by $\widetilde{x}^*$.

Then to generate these scenarios, we measure the distance (proximity) of two scenarios by means of the space of cost values ($\mathbb{R}$).

First, let us first decompose the implementation error as follows:

$$\begin{aligned}\left|f\left(\widetilde{x}^*\right)-v^*\right|=\left|f\left(\widetilde{x}^*\right)-f\left(x^*\right)\right|&=\left|f\left(\widetilde{x}^*\right)-\widetilde{f}\left(\widetilde{x}^*\right)+\widetilde{f}\left(\widetilde{x}^*\right)-f\left(x^*\right)\right|\\&\leq\left|f\left(\widetilde{x}^*\right)-\widetilde{f}\left(\widetilde{x}^*\right)\right|+\left|\widetilde{f}\left(\widetilde{x}^*\right)-f\left(x^*\right)\right|\end{aligned}\tag{2}$$

The second term can be further bounded by:

$$\begin{aligned}\left|\widetilde{f}\left(\widetilde{x}^*\right)-f\left(x^*\right)\right|&=\max\left\{\widetilde{f}\left(\widetilde{x}^*\right)-f\left(x^*\right),f\left(x^*\right)-\widetilde{f}\left(\widetilde{x}^*\right)\right\}\\&\leq\max\left\{\widetilde{f}\left(x^*\right)-f\left(x^*\right),f\left(\widetilde{x}^*\right)-\widetilde{f}\left(\widetilde{x}^*\right)\right\}\\&\leq\max\left\{\left|\widetilde{f}\left(x^*\right)-f\left(x^*\right)\right|,\left|f\left(\widetilde{x}^*\right)-\widetilde{f}\left(\widetilde{x}^*\right)\right|\right\}\\&=\max_{x\in\{x^*,\widetilde{x}^*\}}|\widetilde{f}(x)-f(x)|\end{aligned}\tag{3}$$

Finally, by combining (2) and (3) we can bound the implementation error as follows. Let $\widetilde{X} \subseteq X$ be any feasible set such that $\{x^*, \widetilde{x}^*\} \subset \widetilde{X}$ and we have

$$
\begin{aligned}
|f(\widetilde{x}^*) - v^*| &\leq 2 \max_{x \in \{x^*, \widetilde{x}^*\}} |\tilde{f}(x) - f(x)| \\
&\leq 2 \max_{x \in \widetilde{X}} |\tilde{f}(x) - f(x)| \\
&= 2 \max_{x \in \widetilde{X}} \left| \frac{1}{N} \sum_{i=1}^{N} F(x, \xi_i) - \sum_{k=1}^{K} p_k F\left(x, \widetilde{\xi}_k\right) \right| \\
&= 2 \max_{x \in \widetilde{X}} \left| \frac{1}{N} \sum_{k=1}^{K} \sum_{i \in C_k} F(x, \xi_i) - \sum_{k=1}^{K} \frac{|C_k|}{N} F\left(x, \widetilde{\xi}_k\right) \right| \\
&= 2 \max_{x \in \widetilde{X}} \left| \sum_{k=1}^{K} \frac{|C_k|}{N} \left(\frac{1}{|C_k|} \sum_{i \in C_k} F(x, \xi_i) - F\left(x, \widetilde{\xi}_k\right) \right) \right| \\
&\leq 2 \sum_{k=1}^{K} p_k \sup_{x \in \widetilde{X}} \left| \frac{1}{|C_k|} \sum_{i \in C_k} F(x, \xi_i) - F\left(x, \widetilde{\xi}_k\right) \right| \\
&=: 2 \sum_{k=1}^{K} p_k D(C_k)
\end{aligned}
\tag{4}
$$

The quantity $D(C_k)$ can be seen as the discrepancy of the cluster C_k . It measures how much the cost function $F(x, \widetilde{\xi}_k)$ of its representative scenario $\widetilde{\xi}_k$ matches the average cost values of the whole cluster C_k over the feasible set $\tilde{X}$. we then approximate $D(C_k)$ by:

$$
D(C_k) \simeq \left| \frac{1}{|C_k|} \sum_{i \in C_k} F(x_k^*, \xi_i) - F\left(x_k^*, \widetilde{\xi}_k\right) \right|
$$

where $x_k^* \in \underset{x \in X}{\operatorname{argmin}} F(x, \widetilde{\xi}_k)$. The next subsection describes the algorithm in more details and analyze its computational cost. Then we can state the algorithm as follows:

Algorithm 1.
1. Compute the opportunity-cost matrix $\boldsymbol{V} = (V_{i,j})$ where $V_{i,j} = F(x_i^*, \xi_j)$ and $x_i^* \in \underset{x \in X}{\operatorname{argmin}} F(x, \xi_i)$.
2. Find a partition of the set $\{1, \ldots, N\}$ into K clusters $C_1, \ldots, C_K$ and their representatives $r_1 \in C_1, \ldots, r_K \in C_K$ such that the following clustering discrepancy is minimized:

$$
\sum_{k=1}^{K} p_k \left| V_{r_k, r_k} - \frac{1}{|C_k|} \sum_{j \in C_k} V_{r_k, j} \right|,
$$

where $p_k = \frac{|C_k|}{N}$.

The second step is equivalent to

$$
\begin{array}{lll}
\min & \frac{1}{N} \sum_{i=1}^{N} t_i & \\
\text{s.t.} & t_j \geq \sum_{i=1}^{N} x_{ij} V_{j,i} - \sum_{i=1}^{N} x_{ij} V_{j,j}, & \forall j \in \{1, \ldots, N\}; \\
& t_j \geq \sum_{i=1}^{N} x_{ij} V_{j,j} - \sum_{i=1}^{N} x_{ij} V_{j,i}, & \forall i \in \{1, \ldots, N\}; \\
& x_{ij} \leq u_j, \quad x_{jj} = u_j & \forall (i,j) \in \{1, \ldots, N\}^2; \\
& \sum_{j=1}^{N} x_{ij} = 1, \quad \sum_{j=1}^{N} u_j = K & \forall i \in \{1, \ldots, N\}.
\end{array}
\tag{5}
$$

It should be noted that this means that two scenarios with very different cost values may still be included in the same cluster if there exists a third scenario whose value provides a mid-ground between them.

2 Prerequisite on Gröbner Basis and Graver Basis

For an integer programming problem:

$$(IP)_{c,b} : \min \{cx : Ax = b \text{ and } x \in N^n\},$$

we have Grobner Basis $\mathcal{G}_{c,A}$ and Graver Basis $\mathcal{GR}_A$, where the subscript represents what necessarily determines the basis.

First we recall that the reduced Gröbner Basis and test set are related as follows.

$$(x^{\alpha_i} - x^{\beta_i}) \leftrightarrow (\alpha_i - \beta_i)$$

And by choosing appropriate cost c and monomial order $>$, we can get all monomial orders in the form of the composite order $>_c$, for example, lexicographic order. See for the details in the proof of theorem 10 in [2].

3 Improvement in computation by Gröbner Basis

When ξ_i varies,

$$F(x, \xi_i)$$

also varies. So to best diminish the computational cost, we study the relationships between the Gröbner Basis of $(IP)_{c,b} : \min \{cx : Ax = b \text{ and } x \in N^n\}$ as b, c varies. Generally, there's no relationship between them.

It is worth to be noted that if the matrix A changes when ξ_i varies, we couldn't make use of the relationships between the Gröbner Basis. This is because they are unpredictable.

Example 1 Let

$$A = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

Then we have the toric ideal of A, denoted by $I_A = \{x^u - x^v : u, v \in \mathbb{Z}^n_{\geq 0}, Au = Av\}$, is $\left\langle x-y, y-z, z-x \right\rangle$. So let $>_c =$ lexicographic order, we have $\mathcal{G}_{c,A} = \{(1,0,1),(0,1,1)\}$ and $>_{c'} =$ lexicographic order w.r.t.$z > x > y$ we have $\mathcal{G}_{c',A} = \{(1,1,0),(0,1,1)\} \neq \mathcal{G}_{c,A}$. The results above are proved by the following **Sage Codes**.

```
sage: A=matrix([1,1,1])
sage: IA = ToricIdeal(A)
sage: IA
Ideal (z0 - z1, -z1 + z2) of Multivariate Polynomial Ring in z0, z1, z2 over Rational Field

sage: I = Ideal([x-y,y-z,z-x])
sage: I.groebner_basis()
[x - z, y - z]
```

Actually, no relationships can be found when b, c are varying since the change can be random. However, we can significantly reduce the time of computation by means of the following route.

Algorithm 2. (Compute the Gröbner Basis fixing each ξ_j)
Input: matrix A and c of $(IP)_{c,b}$
Output: a test set $\mathcal{T}_c$ for $(IP)_{c,b}$
1. Compute a small generating set of Ker(A). (**Algorithm 8 in paper 1**)
2. Apply the bunchberger's algorithm to compute the the Gröbner Basis of ξ_j

So from the algorithm, only A and the required ingredient of the scenario ξ_j is needed as an input. And the computational cost is greatly reduced since we don't need to compute the Gröbner Basis of an ideal spanned by a large set any more.

4 Improvement in computation by Graver Basis

First we shall recall some new facts about graver basis of $(IP)_{c,b}$, which is denoted by $\mathcal{GR}_A$.

Proposition 1. Let $\mathcal{G}_{A,c}$ be the reduce Gröbner basis of $(IP)_{c,b}$. Then

$$\bigcup_{c\in\mathbb{Z}^n} \mathcal{G}_{A,c} \subseteq \mathcal{GR}_A.$$

Moreover, for the relationship between the Graver Basis (Universal Gröbner Basis) of the SIP

$$\min\left\{c^T x + \sum_{i=1}^{N} p_i q^T y_i : Ax = b, x \in \mathbb{Z}_+^m, Tx + Wy_i = h_i, y_i \in \mathbb{Z}_+^n, i = 1, \ldots, N\right\} \tag{6}$$

and that of the IP

$$\min\left\{c^T x + p_1 q^T y : Ax = b, x \in \mathbb{Z}_+^m, Tx + Wy = h, y \in \mathbb{Z}_+^n\right\} \tag{7}$$

we assert

Theorem 1. (new) Denote the Graver Basis of (6) by $\mathcal{GR}_N$, and that of (7) by $\mathcal{GR}_1$, $i = 1, 2, ..., N$. Then if A is of full rank (specially if A is a zero matrix), we have

$$\mathcal{GR}_N = \{(0, 0, ..., v_i, ..., 0) : (0, v_i) \in \mathcal{GR}_1, i = 1, 2, ..., n\}$$

and

Theorem 2. (new) Denote all building blocks of Graver Basis of (7) by $\mathcal{H}_N$. Then we have for $N \geq 2$

$$\mathcal{H}_N = \mathcal{H}_2$$

From above theorems and propositions, we have two new algorithms.

Algorithm 3. (Augmentation Algorithm for IP)
Input: a feasible solution z_0 to $(IP)_{c,b}$, a universal test set $\mathcal{T}$ for $(IP)_{c,b}$
Output: an optimal point $z_{\min}$ of $(IP)_{c,b}$
while there is $t \in \mathcal{T}$ with $c^T t > 0$ such that $z_0 - t$ is feasible do

$$z_0 := z_0 - t$$

return:z_0

Algorithm 4. (Construction of Universal Test Set for (6) under the Condition of Theorem 1)
Input: a universal test set $\mathcal{T}$ for (7)
Output: a universal test set $\mathcal{T}$ for (6)
1. Construct the set as in Theorem 1.

Algorithm 5. (Augmentation Algorithm for (6))
Input: a feasible solution z_0 to (7) , a universal test set $\mathcal{T}$ for (6)
Output: an optimal point $z_{\min}$ of $(IP)_{c,b}$
1. Compute the building blocks for (6) when $N = 2$.
2. Apply building blocks to find an optimum of (6) (**Lemma 9 and Lemma 10 in Paper 1**)

References

[1] Keutchayan, Julien, et al. “Problem-Driven Scenario Clustering in Stochastic Optimization.” ArXiv.org, 22 June 2021, www.arxiv.org/abs/2106.11717.

[2] Yuchen Ge. “Algebra for Machine Learning and Stochastic Programming.”August, https://gycdwwd.github.io/Writing/Algsto.pdf.