Difference of convex algorithms for bilevel programs with applications in hyperparameter selection

Jane Ye

University of Victoria, Canada

Lecture 4 at the Forum on Developments and Origins of Operations Research
November 27, 2021
Organizers: The Mathematical Programming Branch of OR Society of China
& Southern University of Science and Technology

Outline

- Introduction to applications of bilevel programs in hyperparameter selection
- Difference of convex algorithms for difference of convex program
- Difference of convex algorithms for bilevel programs with applications to support vector classification.

Bilevel programs

Consider bilevel program:

$$(BP) \quad \min_{x \in X} \quad F(x, y)$$
$$s.t. \quad y \in S(x)$$

where $S(x)$ denotes the set of optimal solutions of the lower level problem:

$$(P_x) \quad \min_{y \in Y} \quad f(x, y),$$
$$s.t. \quad g(x, y) \leq 0.$$

Here the defining functions may be nonsmooth.

Applications in machine learning

- It was first introduced to the model selection in machine learning by Bennett, Hu, Ji, Kunapuli and Pang in 2006.
- One of the main tasks of Machine Learning (ML) is, from given data, to design a model which can predict the future. Most of ML models have parameters that need to be prefixed. Such parameters are called hyperparameters. Prediction performance of ML models significantly relies on the choice of hyperparameters. It has been recognized that this matter is one of the most crucial ones in ML.
- Recently there are more and more works on hyperparameter optimization and meta-learning via bilevel optimization. Moore et al. (Mach Learn 2011); Franceschi et al. (ICML, 2017); Franceschi et al. (ICML, 2018);Okuno and Takada (2018); MacKay et al.(2019);Rajeswaran et al. (2019); Zügner and Günnemann (2019); Okuno and Kawana (2020); etc.

Model selection

- Let $\mathbf{a} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ be predictor and the response variables, respectively. Suppose we have a data set containing ℓ observations $\Omega := \{(\mathbf{a}_1, b_1), \ldots, (\mathbf{a}_\ell, b_\ell)\}$. We try to fit a statistical model to study the relationship between $\mathbf{a}$ and b. Assuming $b \approx \mathbf{a}^T\theta$, we try to estimate θ.
- If $n \geq \ell$, i.e., the number of predictor variables are larger than the number of samples, the classical linear regression problem is ill-post. Some irrelevant variables may be included in the fitted model.
- Using lasso (Tibshirani 1996), for given $\lambda > 0$ the regularized problem is solved:

$$\min_{\theta} \sum_{(\mathbf{a}_j, b_j) \in \Omega} (\mathbf{a}_j^T\theta - b_j)^2 + \lambda \|\theta\|_1.$$

  Bigger λ encourage sparser optimal solution for θ. But how to select λ so that the model is correct?
- The selection of λ is often performed via T-fold cross validation.

T-fold cross validation

Given a data set: $\Omega := \{(\mathbf{a}_1, b_1), \ldots, (\mathbf{a}_\ell, b_\ell)\}$.
Step 1: Randomly split the data set into T (e.g., $T = 3, 5, 10$) disjoint blocks with approximately equal size:

$$\Omega = \Omega_1 \cup \cdots \cup \Omega_T.$$

Step 2: For $t = 1, \ldots, T$, use $\Omega^t_{\text{val}} = \Omega_t$ as the test set and the rest $(T-1)$ blocks as the training set Ω^t_{trn}, and compute the fitted value $\theta_t \in \arg\min_\theta \sum_{(a_j, b_j) \in \Omega^t_{\text{trn}}} (\mathbf{a}_j^T \theta - b_j)^2 + \lambda \|\theta\|_1$.
Step 3, Compute the validation mean-squared-error on the observations in Ω^t_{val}, i.e., $\text{MSE}_t(\theta) := \sum_{(\mathbf{a}_j, b_j) \in \Omega^t_{\text{val}}} (\mathbf{a}_j^T \theta - b_j)^2$, and compute the cross validation error

$$CV(\theta_1, \ldots, \theta_T) := \frac{1}{T} \sum_{t=1}^{T} \text{MSE}_t(\theta_t).$$

Step 4. Repeat Steps 2 and 3 for various values of $\lambda > 0$.
Step 5. Find λ^* that minimize the cross validation error and in the mean time θ^* the best fitted value.

Cross validation as a bilevel program

- In statistics, either a grid search or a path following algorithm is performed on λ values to select the value of λ for which the cross-validation error is smallest. But these approaches do not scale well and have a lot of limitations.
- In essence the cross validation in lasso is the following bilevel program:

$$\min_{\lambda,\theta_1,\ldots,\theta_T} \quad \frac{1}{T}\sum_{t=1}^{T} \mathrm{MSE}_t(\theta_t)$$
$$\lambda > 0 \text{ and for each } t = 1,\ldots,T$$
$$\theta_t \in \arg\min_{\theta} \sum_{(\mathbf{a}_j,b_j)\in\Omega^t_{trn}} (\mathbf{a}_j^T\theta - b_j)^2 + \lambda\|\theta\|_1$$

- If the above bilevel program can be solved, then we can obtain the optimal penalty parameter λ^* and the best fitted value θ^* at once!

Nonsmooth bilevel programs

Question: How do we solve a nonsmooth bilevel program?
Almost all algorithms require the smoothness of defining functions. Sometimes non-smoothness can be dealt with by introducing auxiliary variables and constraints to reformulate a nonsmooth lower level program as a smooth one but then the number of variables or constraints would increase. Moreover, MPEC approach is not reliable due to the extra variables from Lagrange multipliers.

Convexity of the value function

We say that the the lower level program

$$(P_x) \qquad \min_{y \in Y} f(x,y) \quad \text{s.t. } g(x,y) \leq 0$$

is completely convex if all functions $f(x,y)$ and $g(x,y)$ are convex in both variables x and y and Y is a convex set. In this case the value function is convex and the value function constraint becomes a difference of convex (DC) constraint:

$$f(x,y) - V(x) \leq 0.$$

Difference of convex optimization

Let g and h are convex functions. Consider DC program:

$$\min f(x) := g(x) - h(x).$$

Let x^k be given. Take $\xi^k \in \partial h(x^k)$. Then by convexity of h,

$$f(x) \leq g(x) \qquad \underbrace{-h(x^k) - \langle \xi^k, x - x^k \rangle}_{\text{linearization of the concave part}}. \tag{1}$$

$$\text{Solve } x^{k+1} \in \arg\min_x \{g(x) - \langle \xi,^k x - x^k \rangle\}. \tag{2}$$

$$\begin{aligned} f(x^{k+1}) &\leq g(x^{k+1}) - h(x^k) - \langle \xi^k, x^{k+1} - x^k \rangle \quad \text{by majorization (1)} \\ &\leq g(x^k) - h(x^k) \quad \text{by mimization (2)} \\ &= f(x^k). \end{aligned}$$

Hence the value of f decreases monotonically in each iteration!

Difference of Convex Algorithms (DCA)

Many functions can be represented as a difference of convex (DC) functions: lower C^2 functions and C^{1+} functions are DC functions, and the class of DC functions is closed under many operations. The difference of convex algorithm (DCA) (cf. review paper by Horst and Thoai 1999) can be used to solve a DC program:

$$(\mathrm{DC}) \qquad \begin{array}{ll} \min\limits_{z\in\Sigma} & f_0(z) := g_0(z) - h_0(z) \\ s.t. & f_1(z) := g_1(z) - h_1(z) \le 0, \end{array}$$

where Σ is a closed convex subset of $\mathbb{R}^d$ and $g_0(z), h_0(z), g_1(z), h_1(z) : \Sigma \to \mathbb{R}$ are convex functions. DCA linearizes the concave part of the DC function. DCA converges to a KKT point provided that

- all functions are convex and Lipschitz continuous.
- the extended MFCQ (EMFCQ) holds.

Definition

Let $\bar{z}$ be a feasible solution of problem (DC). We say that $\bar{z}$ is a stationary/KKT point of problem (DC) if there exists a multiplier $\lambda \geq 0$ such that

$$0 \in \partial g_0(\bar{z}) - \partial h_0(\bar{z}) + \lambda(\partial g_1(\bar{z}) - \partial h_1(\bar{z})) + \mathcal{N}_\Sigma(\bar{z}),$$
$$(g_1(\bar{z}) - h_1(\bar{z}))\lambda = 0.$$

Definition

Let $\bar{z}$ be a feasible point of problem (DC). We say that NNAMCQ/MFCQ holds at $\bar{z}$ for problem (DC) if either $f_1(\bar{z}) < 0$ or $f_1(\bar{z}) = 0$ but

$$0 \notin \partial g_1(\bar{z}) - \partial h_1(\bar{z}) + \mathcal{N}_\Sigma(\bar{z}). \tag{3}$$

Let $\bar{z} \in \Sigma$, we say that ENNAMCQ/EMFCQ holds at $\bar{z}$ for problem (DC) if either $f_1(\bar{z}) < 0$ or $f_1(\bar{z}) \geq 0$ but (3) holds.

Proposition

Let $\bar{z}$ be a local solution of problem (DC). If NNAMCQ/MFCQ holds at $\bar{z}$ and all functions g_0, g_1, h_0, h_1 are Lipschitz around point $\bar{z}$, then $\bar{z}$ is a KKT point of problem (DC).

inexact proximal difference of convex algorithm (iPDCA)

- Given a current iterate $z^k \in \Sigma$, select a subgradent $\xi_i^k \in \partial h_i(z^k), i = 1, 2.$
- Compute z^{k+1} as an approximate minimizer of the following strongly convex subproblem

$$\min_{z\in\Sigma} \ \tilde{\varphi}_k(z) := \ g_0(z) \ \underbrace{-h_0(z^k) - \langle \xi_0^k, z - z^k\rangle}_{\text{linearization of } -h_0(z) \text{ at } z^k}$$
$$+ \beta_k \max\{g_1(z) \ \underbrace{-h_1(z^k) - \langle \xi_1^k, z - z^k\rangle}_{\text{linearization of } -h_1(z) \text{ at } z^k} \ , 0\} + \frac{\rho}{2}\|z - z^k\|^2,$$

where β_k is a penalty parameter and $\rho > 0$ is a given constant.

Inexact conditions for subproblems of iPDCA and the rule for the penalty parameter update

Note that z^{k+1} is an optimal solution if and only if

$$0 \in \partial\tilde{\varphi}_k(z^{k+1}) + \mathcal{N}_\Sigma(z^{k+1}).$$

Condition 1: $\mathrm{dist}(0, \partial\tilde{\varphi}_k(z^{k+1}) + \mathcal{N}_\Sigma(z^{k+1})) \le \zeta_k$, for $\zeta_k \ge 0$: $\sum_{k=0}^{\infty} \zeta_k^2 < \infty$,

Condition 2:
$\mathrm{dist}(0, \partial\tilde{\varphi}_k(z^{k+1}) + \mathcal{N}_\Sigma(z^{k+1})) \le \frac{\sqrt{2}}{2}\rho\|z^k - z^{k-1}\|$.

Update parameter β_{k+1} by the rule:

$$\beta_{k+1} = \begin{cases} \beta_k + \delta_\beta, & \text{if } \max\{\beta_k, 1/t^{k+1}\} < \|z^{k+1} - z^k\|^{-1}, \\ \beta_k, & \text{otherwise.} \end{cases}$$

$t^{k+1} := \max\{g_1(z^{k+1}) - h_1(z^k) - \langle \xi_1^k, z^{k+1} - z^k \rangle, 0\}$.

Algorithm 1 iP-DCA

1: Take an initial point $z^0 \in \Sigma$; $\delta_\beta > 0$; an initial penalty parameter $\beta_0 > 0$, $tol > 0$.

2: **for** $k = 0, 1, \ldots$ **do**

1. Compute $\xi_i^k \in \partial h_i(z^k)$, $i = 0, 1$.
2. Obtain an inexact solution z^{k+1} of the subproblem.
3. Stopping test. Compute
$t^{k+1} := \max\{g_1(z^{k+1}) - h_1(z^k) - \langle \xi_1^k, z^{k+1} - z^k \rangle, 0\}$.
Stop if $\max\{\|z^{k+1} - z^k\|, t^{k+1}\} < tol$.
4. Penalty parameter update. Set

$$\beta_{k+1} = \begin{cases} \beta_k + \delta_\beta, & \text{if } \max\{\beta_k, 1/t^{k+1}\} < \|z^{k+1} - z^k\|^{-1}, \\ \beta_k, & \text{otherwise.} \end{cases}$$

5. Set $k := k + 1$.

3: **end for**

Theorem

Suppose f_0 is bounded below on Σ and the sequences $\{z^k\}$ and $\{\beta_k\}$ generated by iP-DCA are bounded. Moreover suppose functions g_0, g_1, h_1, h_0 are locally Lipschitz on set Σ. Then every accumulation point of $\{z^k\}$ is a KKT point of problem (DC).

Proposition

Suppose that the iterate sequence $\{z^k\}$ generated by iP-DCA is bounded. Moreover suppose functions g_0, g_1, h_1, h_0 are Lipschitz around at any accumulation point of $\{z^k\}$. Assume that ENNAMCQ/EMFCQ holds at any accumulation points of the sequence $\{z^k\}$. Then the sequence $\{\beta_k\}$ must be bounded.

Difference of convex bilevel program

$$\begin{aligned} \min_{x,y} \quad & F(x,y) := F_1(x,y) - F_2(x,y) \\ s.t. \quad & x \in X, y \in S(x) := \arg\min_{y \in Y} \{ f(x,y) \ \text{ s.t.} g(x,y) \leq 0 \}, \end{aligned}$$

where $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^m$ are nonempty closed convex sets, $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l$ is convex on an open convex set containing the set $X \times Y$, and the functions $F_1, F_2, f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ are convex on an open convex set containing the set

$$C := \{(x,y) \in X \times Y : g(x,y) \leq 0\}.$$

By Lampariello and Sagratella (2020), if $f(x,y) = f_1(x,y_1) + f_2(y_2)$ where f_2 is convex, $f_1(\cdot, y_1)$ is convex for every y_1 and $f_1(x,\cdot)$ is uniformly strongly convex for every x, then by adding a term $\frac{\beta}{2} x^T x$ with large β to $f(x,y)$, the lower level problem can be reformulated as one with a completely convex objective.

Standing Assumptions

(I) $S(x) \neq \emptyset$ for all $x \in X$. For all x in an open convex set $\mathcal{O} \supseteq X$, the feasible region $\mathcal{F}(x) := \{y \in Y : g(x,y) \leq 0\}$ is nonempty and $f(x,y)$ is bounded below on $\mathcal{F}(x)$.

(II) Assume that the partial derivative formula holds for each of the lower level objective and constraint functions:

$$\partial\phi(x,y) = \partial_x\phi(x,y) \times \partial_y\phi(x,y).$$

Some sufficient conditions for the partial derivative formula:

- $\phi(x,y) = \phi_1(x) + \phi_2(y)$.
- $\phi(x,y)$ is C^1 respect to either x or y.

lasso problem as a bilevel program with a completely convex lower level program

By change of variable $r := \frac{1}{\lambda}$, lasso problem can be equivalently reformulated as:

$$\min_{r,\theta_1,\ldots,\theta_T} \quad CV(\theta_1,\ldots,\theta_T)$$

$$r > 0 \text{ and for each } t = 1,\ldots,T$$

$$\theta_t \in \arg\min_{\theta} \sum_{(a_j,b_j)\in\Omega^t_{trn}} \frac{(\mathbf{a}_j^T\theta - b_j)^2}{r} + \|\theta\|_1.$$

Since a square over linear function

$$\phi(\mathbf{x}, r) = \|\mathbf{x}\|^2/r$$

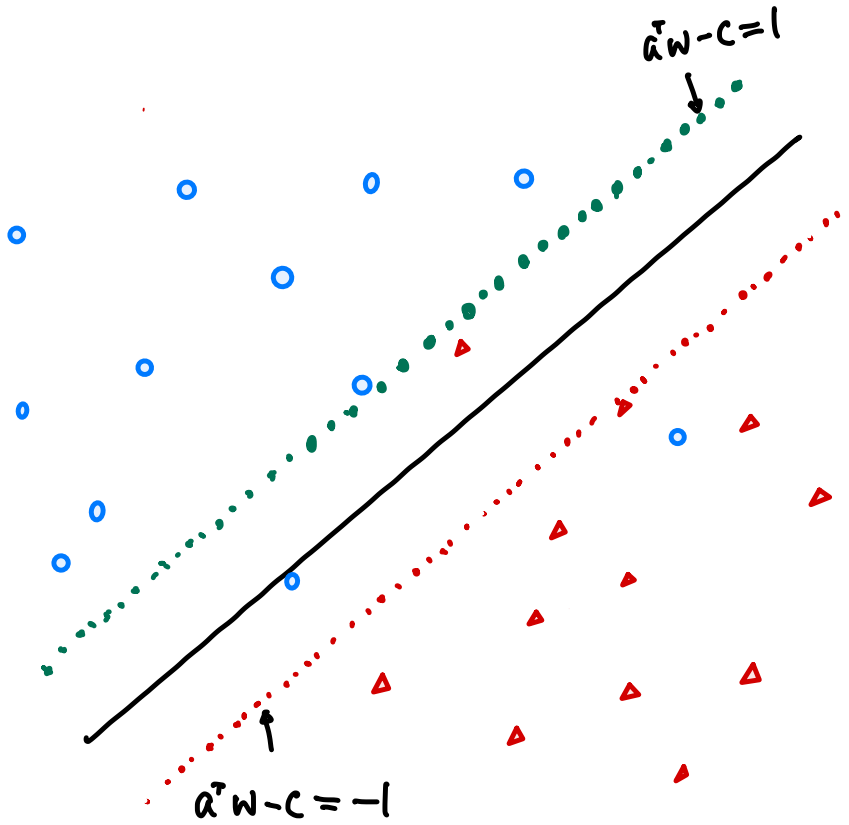
is completely convex, the lower level is a completely convex bilevel program.

Support vector (SV) classification

Consider the problem of separating the set of labeled training vectors belonging to two separate classes. Given a data set $\Omega := \{(\mathbf{a}_j, b_j)\}_{j=1}^{\ell}$ where $\mathbf{a}_j \in \mathbb{R}^n$, with $b_j = \pm 1$ indicating the class membership using a hyperplane, $\mathbf{a}^T\mathbf{w} - c = 0$.
Given $\lambda > 0$, SV classification is to solve

$$\min_{\mathbf{w},\, c} \left\{ \underbrace{\sum_{(a_j,b_j)\in\Omega} \max(1 - b_j(\mathbf{a}_j^T\mathbf{w} - c), 0)}_{\text{classification error}} + \frac{\lambda}{2} \underbrace{\|\mathbf{w}\|^2}_{\text{margin error}} \right\}.$$

- SV classification is to minimize the trade off of the number of misclassified points and the size of the margin.
- We can also add the box constraints to w, i.e., $-\bar{\mathbf{w}} \le \mathbf{w} \le \bar{\mathbf{w}}$.



- Define parallel planes
- minimize points on the wrong side
- maximize margin of separation $\frac{2}{\|w\|}$

The bilevel model for support vector (SV) classification

Given a training set $\Omega := \{(\mathbf{a}_j, b_j)\}_{j=1}^{\ell}$ where $\mathbf{a}_j \in \mathbb{R}^n$, and the labels $b_j = \pm 1$ indicate the class membership. The bilevel model for SV classification using T-fold cross validation (Kunapuli, Bennett, Hu and Pang, 2008):

$$\min_{\lambda, \bar{\mathbf{w}}, \mathbf{w}^1, \ldots \mathbf{w}^T, \mathbf{c}} \quad \Theta(\mathbf{w}^1, \ldots, \mathbf{w}^T, \mathbf{c})$$

$$:= \frac{1}{T} \sum_{t=1}^{T} \sum_{(\mathbf{a}_j, b_j) \in \Omega^t_{\mathbf{val}}} \max(1 - b_j(\mathbf{a}_j^T \mathbf{w}^t - c^t), 0)$$

$$s.t. \quad \lambda_{lb} \leq \lambda \leq \lambda_{ub}, \quad \bar{\mathbf{w}}_{lb} \leq \bar{\mathbf{w}} \leq \bar{\mathbf{w}}_{ub}, \text{and for } t = 1, \ldots, T :$$

$$(\mathbf{w}^t, c^t) \in \underset{\substack{-\bar{\mathbf{w}} \leq \mathbf{w} \leq \bar{\mathbf{w}} \\ c \in \mathbb{R}}}{\operatorname{argmin}} \left\{ \sum_{(\mathbf{a}_j, b_j) \in \Omega^t_{trn}} \max(1 - b_j(\mathbf{a}_j^T \mathbf{w} - c), 0) + \frac{\lambda}{2} \|\mathbf{w}\|^2 \right\}.$$

The bilevel model for SV classification (Kunapuli, Bennett, Hu and Pang, 2008)

By changing the variable λ to $\mu := \frac{1}{\lambda}$ we can reformulate the above SV bilevel model selection equivalently as the following bilevel program with a completely convex lower level program

$$
\begin{aligned}
&\min_{\mu,\bar{\mathbf{w}},\mathbf{w}^1,\ldots,\mathbf{w}^T,\mathbf{c}} \quad \Theta(\mathbf{w}^1,\ldots,\mathbf{w}^T,\mathbf{c}) \\
(\text{SVBP}) \qquad &s.t. \quad \frac{1}{\lambda_{ub}} \le \mu \le \frac{1}{\lambda_{lb}}, \quad \bar{\mathbf{w}}_{lb} \le \bar{\mathbf{w}} \le \bar{\mathbf{w}}_{ub}, \\
&\qquad (\mathbf{w}^1,\ldots,\mathbf{w}^T,\mathbf{c}) \in S(\mu,\bar{\mathbf{w}}),
\end{aligned}
$$

where $S(\mu,\bar{\mathbf{w}})$ is the set of optimal solutions of the lower level problem

$$
(P_{\mu,\bar{\mathbf{w}}}) \qquad \min_{\substack{-\bar{\mathbf{w}} \le \mathbf{w}^t \le \bar{\mathbf{w}} \\ c^t \in \mathbb{R} \\ t=1,\ldots,T}} \left\{ \sum_{t=1}^{T} \left(\frac{\|\mathbf{w}^t\|^2}{2\mu} + \sum_{j \in \Omega^t_{trn}} \max(1 - b_j(\mathbf{a}_j^T \mathbf{w}^t - c^t), 0) \right) \right\}
$$

Reformulation of the bilevel program as a DC program

- If both f and g are completely convex, and X, Y are convex, then the value function of the lower level program

$$V(x) := \inf_{y \in Y} \{f(x,y) \mid g(x,y) \leq 0\},$$

must be convex.

- The equivalent reformulation of the DC bilevel program from the value function approach:

$$(\mathrm{VP}) \qquad \begin{aligned} \min_{(x,y)\in C} \quad & F_1(x,y) - F_2(x,y) \\ s.t. \quad & f(x,y) - V(x) \leq 0, \end{aligned}$$

where $C := \{(x,y) \in X \times Y \mid g(x,y) \leq 0\}$ is a DC program.

Difficulties

Consider the DC bilevel program in the form:

$$(\mathrm{VP}) \qquad \begin{array}{ll} \min\limits_{(x,y)\in C} & F_1(x,y) - F_2(x,y) \\ s.t. & f(x,y) - V(x) \leq 0, \end{array}$$

where $C := \{(x,y) \in X \times Y \mid g(x,y) \leq 0\}$ To apply the difference of convex algorithm (DCA), cf. review paper by Horst and Thoai 1999, we need to answer the following two questions.

(a) Is the value function convex and locally Lipschitz on the convex set X and how to obtain one element from $\partial V(x)$ in terms of problem data?

(b) Under what condition, the constraint qualification EMFCQ holds?

Lipschitz continuity and the subdifferential of the value function

Using the convex analysis in Rockafellar (1970), we can obtain:

- Under Standing assumption (I), all functions F_1, F_2, f and g are convex and locally Lipschitz continuous, and the value function $V(x)$ is convex and locally Lipschitz continuous on X;
- For any $x \in X$ and $y \in S(x)$

$$\bigcup_{\gamma \in KT(x,y)} \left(\partial_x f(x,y) + \sum_{i=1}^{l} \gamma_i \partial_x g_i(x,y) \right) \subseteq \partial V(x),$$

where $KT(x,y)$ denotes the set of KKT multipliers of the lower-level problem (P_x),

$$KT(x,y) := \left\{ \gamma \in \mathbb{R}^l_+ \middle| 0 \in \partial_y f(x,y) + \partial_y g(x,y)^T \gamma + \mathcal{N}_Y(y), \quad g(x,y)^T \gamma = 0 \right\}.$$

Motivations for studying the approximate bilevel program

- Due to the value function constraint, (VP) violates MFCQ/NNAMCQ at each feasible point.
- To deal with this issue, we consider the following approximate DC bilevel program for $\epsilon \geq 0$,

$$(\mathrm{VP})_\epsilon \qquad \min_{(x,y)\in C} \quad F_1(x,y) - F_2(x,y)$$
$$s.t. \quad f(x,y) - V(x) \leq \epsilon.$$

- The solutions of $(\mathrm{VP})_\epsilon$ approximate a true solution of the original bilevel program as ϵ approaches zero (Lin, Xu and JY (2014)).
- For any $\epsilon > 0$, we can prove that the approximate program $(\mathrm{VP})_\epsilon$ always satisfies EMFCQ/ENNAMCQ on

$$C := \{(x,y) \in X \times Y : g(x,y) \leq 0\}.$$

EMFCQ/ENNAMCQ for $(\mathrm{VP})_\epsilon$ with $\epsilon > 0$

Definition

Let $(\bar{x}, \bar{y}) \in C$. We say that NNAMCQ holds at $(\bar{x}, \bar{y})$ for problem $(\mathrm{VP})_\epsilon$ if either $f(\bar{x}, \bar{y}) - V(\bar{x}) < \epsilon$ or $f(\bar{x}, \bar{y}) - V(\bar{x}) = \epsilon$ but

$$0 \notin \partial f(\bar{x}, \bar{y}) - \partial V(\bar{x}) \times \{0\} + \mathcal{N}_C(\bar{x}, \bar{y}).$$

$(\mathrm{VP})_\epsilon$ with $\epsilon > 0$ satisfies EMFCQ at $(\bar{x}, \bar{y})$

Proof. If $f(\bar{x}, \bar{y}) - v(\bar{x}) < \epsilon$ holds, then EMFCQ holds at $(\bar{x}, \bar{y})$. Now suppose that $f(\bar{x}, \bar{y}) - v(\bar{x}) \geq \epsilon$ and EMFCQ does not hold, i.e.,

$$0 \in \partial f(\bar{x}, \bar{y}) - \partial V(\bar{x}) \times \{0\} + \mathcal{N}_C(\bar{x}, \bar{y}).$$

It follows from the partial subdifferentiation formula that

$$0 \in \begin{bmatrix} \partial_x f(\bar{x}, \bar{y}) - \partial V(\bar{x}) \\ \partial_y f(\bar{x}, \bar{y}) \end{bmatrix} + \mathcal{N}_C(\bar{x}, \bar{y}). \tag{4}$$

$$\mathcal{N}_C(\bar{x}, \bar{y}) = \partial \delta_C(\bar{x}, \bar{y}) \subseteq \partial_x \delta_C(\bar{x}, \bar{y}) \times \partial_y \delta_C(\bar{x}, \bar{y}) \subseteq \mathbb{R}^n \times \mathcal{N}_{C(\bar{x})}(\bar{y}),$$

where $C(\bar{x}) := \{y \in Y \mid g_i(\bar{x}, y) \leq 0, i = 1, \ldots, l\}$. Thus, it follows from (4) that

$$0 \in \partial_y f(\bar{x}, \bar{y}) + \mathcal{N}_{C(\bar{x})}(\bar{y}),$$

which further implies that $\bar{y} \in \mathcal{S}(\bar{x})$. This contradicts to the assumption that $f(\bar{x}, \bar{y}) - v(\bar{x}) \geq \epsilon > 0$.

Inexact proximal difference of convex algorithm (iPDCA)

- Given a current iteration point (x^k, y^k), solve the lower level problem (P_{x^k}) with a global minimizer $\tilde{y}^k$ and a corresponding multiplier denoted by λ^k.
- Select

$$\xi_0^k \in \partial F_2(x^k, y^k), \xi_1^k \in \partial_x f(x^k, \tilde{y}^k) + \partial_x g(x^k, \tilde{y}^k)^T \lambda^k \subseteq \partial V(x^k).$$

- Compute (x^{k+1}, y^{k+1}) as an approximate minimizer of the strongly convex subproblem for $(\mathrm{VP})_\epsilon$ given by

$$\min_{(x,y)\in C} \; F_1(x,y) \underbrace{- F_2(x^k, y^k) - \langle \xi_0^k, (x,y) \rangle}_{\text{linearization of } -F_2 \text{ at } (x^k, y^k)} + \frac{\rho}{2}\|(x,y) - (x^k, y^k)\|^2$$
$$+ \beta_k \max\{f(x,y) \underbrace{- f(x^k, \tilde{y}^k) - \langle \xi_1^k, x - x^k \rangle}_{\text{linearization of } -V(x) \text{ at } x^k} - \epsilon, 0\}.$$

- Update penalty parameter β_{k+1}.

Convergence theorem

Definition

We say a point $(\bar{x}, \bar{y})$ is a KKT point of problem $(\mathrm{VP})_\epsilon$ with $\epsilon \geq 0$ if there exists $\mu \geq 0$ such that

$$\begin{aligned} 0 \in \partial F_1(\bar{x}, \bar{y}) - \partial F_2(\bar{x}, \bar{y}) + \mu(\partial f(\bar{x}, \bar{y}) - \partial V(\bar{x}) \times \{0\}) \\ + \mathcal{N}_C(\bar{x}, \bar{y}), \\ f(\bar{x}, \bar{y}) - V(\bar{x}) - \epsilon \leq 0, \quad \mu\,(f(\bar{x}, \bar{y}) - V(\bar{x}) - \epsilon) = 0. \end{aligned}$$

Theorem

Assume that the upper level objective F is bounded below on C. Let $\{(x^k, y^k)\}$ be an iteration sequence generated by iPDCA . Moreover assume that $KT(x^k, y) \neq \emptyset$ for all $y \in S(x^k)$. Suppose that either $\epsilon > 0$ or $\epsilon = 0$ and the penalty sequence $\{\beta_k\}$ is bounded. Then any accumulation point of $\{(x^k, y^k)\}$ is an KKT point of problem $(\mathrm{VP})_\epsilon$.

Numerical experiments on SV bilevel model selection

We conduct numerical experiments on the SV bilevel model selection problem (SVBP).

$$\min_{\mu,\bar{\mathbf{w}},\mathbf{w}^1,\ldots,\mathbf{w}^T,\mathbf{c}} \quad \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\Omega^t_{val}|}\sum_{j\in\Omega^t_{val}}\max(1-b_j(\mathbf{a}_j^T\mathbf{w}^t_{\lambda,\bar{\mathbf{w}}}-c^t_{\lambda,\bar{\mathbf{w}}}),0)$$

$$s.t. \quad \frac{1}{\lambda_{ub}}\le\mu\le\frac{1}{\lambda_{lb}}, \quad \bar{\mathbf{w}}_{lb}\le\bar{\mathbf{w}}\le\bar{\mathbf{w}}_{ub},$$

$$(\mathbf{w}^1,\ldots,\mathbf{w}^T,\mathbf{c})\in S(\mu,\bar{\mathbf{w}}),$$

where $S(\mu,\bar{\mathbf{w}})$ is the set of optimal solutions of the lower level problem

$$(P_{\mu,\bar{\mathbf{w}}}) \qquad \min_{\substack{-\bar{\mathbf{w}}\le\mathbf{w}^t\le\bar{\mathbf{w}}\\ c^t\in\mathbb{R}\\ t=1,\ldots,T}} \left\{\sum_{t=1}^{T}\left(\frac{\|\mathbf{w}^t\|^2}{2\mu}+\sum_{j\in\Omega^t_{trn}}\max(1-b_j(\mathbf{a}_j^T\mathbf{w}^t-c^t),0)\right)\right\}$$

Numerical experiments on SV bilevel model selection

- Given current iterate $x^k := (\mu^k, \bar{\mathbf{w}}^k)$, solve $(P_{\mu^k, \bar{\mathbf{w}}^k})$ and obtain a solution $\tilde{y}^k := (\tilde{\mathbf{w}}^1, \ldots, \tilde{\mathbf{w}}^T, \tilde{\mathbf{c}}) \in S(x^k)$ and a corresponding KKT multiplier

$$(\gamma^k_{1,1}, \ldots, \gamma^k_{1,T}, \gamma^k_{2,1}, \ldots, \gamma^k_{2,T}) \in KT(x^k, \tilde{y}^k).$$

- Since $F(x, y)$ is convex, we have $\xi^k_0 = 0$. Since both $f(x, y)$ and $g(x, y)$ are smooth in variable $x := (\mu, \bar{\mathbf{w}})$, $\xi^k_1 \in \partial V(x^k)$ can be calculated by

$$\xi^k_1 = \begin{pmatrix} -\frac{\sum_{t=1}^T \|\tilde{\mathbf{w}}^t\|^2}{2(\mu^k)^2} \\ -\sum_{t=1}^T \gamma^k_{1,t} - \sum_{t=1}^T \gamma^k_{2,t} \end{pmatrix}$$

Numerical experiments on SV bilevel model selection

Table: Description of datasets used

Dataset	ℓ_{train}	ℓ_{test}	n	T
australian_scale	345	345	14	3
breast-cancer_scale	339	344	10	3
diabetes_scale	384	384	8	3
mushrooms	4062	4062	112	3
phishing	5526	5529	68	3

The numbers of the upper level variables = the number of hyperparameters for the datasets are $n+1$. The numbers of the lower level variables $= 3(n+1)$.

Numerical experiments on SV bilevel model selection

Table: Numerical results comparing iP-DCA and MPEC approach

Dataset	Method	CV error	Test error	Time(sec)
australian_scale	iP-DCA($\epsilon = 0$, $tol = 10^{-2}$)	0.28 ± 0.03	0.15 ± 0.01	73.7 ± 106.6
	iP-DCA($\epsilon = 0$, $tol = 10^{-3}$)	0.28 ± 0.03	0.15 ± 0.01	81.2 ± 110.8
	iP-DCA($\epsilon = 10^{-2}$, $tol = 10^{-2}$)	0.28 ± 0.03	0.15 ± 0.01	10.7 ± 6.3
	iP-DCA($\epsilon = 10^{-2}$, $tol = 10^{-3}$)	0.28 ± 0.03	0.15 ± 0.01	128.7 ± 74.4
	iP-DCA($\epsilon = 10^{-4}$, $tol = 10^{-2}$)	0.28 ± 0.03	0.15 ± 0.01	74.2 ± 123.8
	iP-DCA($\epsilon = 10^{-4}$, $tol = 10^{-3}$)	0.28 ± 0.03	0.15 ± 0.01	109.0 ± 141.0
	MPEC approach	0.29 ± 0.04	0.15 ± 0.01	491.2 ± 245.1
breast-cancer_scale	iP-DCA($\epsilon = 0$, $tol = 10^{-2}$)	0.06 ± 0.01	0.04 ± 0.00	53.1 ± 67.2
	iP-DCA($\epsilon = 0$, $tol = 10^{-3}$)	0.06 ± 0.01	0.04 ± 0.00	78.3 ± 73.9
	iP-DCA($\epsilon = 10^{-2}$, $tol = 10^{-2}$)	0.06 ± 0.01	0.04 ± 0.00	15.5 ± 2.1
	iP-DCA($\epsilon = 10^{-2}$, $tol = 10^{-3}$)	0.06 ± 0.01	0.04 ± 0.00	108.9 ± 40.4
	iP-DCA($\epsilon = 10^{-4}$, $tol = 10^{-2}$)	0.06 ± 0.01	0.04 ± 0.01	24.6 ± 17.5
	iP-DCA($\epsilon = 10^{-4}$, $tol = 10^{-3}$)	0.06 ± 0.01	0.04 ± 0.01	86.8 ± 59.3
	MPEC approach	0.08 ± 0.01	0.04 ± 0.01	294.5 ± 98.2
diabetes_scale	iP-DCA($\epsilon = 0$, $tol = 10^{-2}$)	0.56 ± 0.03	0.24 ± 0.02	12.0 ± 13.6
	iP-DCA($\epsilon = 0$, $tol = 10^{-3}$)	0.56 ± 0.03	0.24 ± 0.02	25.9 ± 33.2
	iP-DCA($\epsilon = 10^{-2}$, $tol = 10^{-2}$)	0.57 ± 0.03	0.24 ± 0.02	3.1 ± 0.6
	iP-DCA($\epsilon = 10^{-2}$, $tol = 10^{-3}$)	0.56 ± 0.03	0.24 ± 0.02	62.1 ± 31.7
	iP-DCA($\epsilon = 10^{-4}$, $tol = 10^{-2}$)	0.56 ± 0.03	0.24 ± 0.02	12.7 ± 19.7
	iP-DCA($\epsilon = 10^{-4}$, $tol = 10^{-3}$)	0.56 ± 0.03	0.24 ± 0.02	39.2 ± 45.7
	MPEC approach	0.59 ± 0.03	0.25 ± 0.02	346.7 ± 216.9

Numerical experiments on SV bilevel model selection

Table: Numerical results of iP-DCA on datasets "mushrooms" and "phishing" with $tol = 10^{-2}$

Dataset	Method	CV error	Test error	Time(sec)
mushrooms	iP-DCA($\epsilon = 0$)	6.36e-04 $\pm$ 5.94e-04	0 $\pm$ 0	334.3 $\pm$ 346.1
	iP-DCA($\epsilon = 10^{-2}$)	1.53e-03 $\pm$ 3.85e-03	3.57e-04 $\pm$ 1.34e-03	109.3 $\pm$ 35.2
	iP-DCA($\epsilon = 10^{-4}$)	6.38e-04 $\pm$ 6.08e-04	0 $\pm$ 0	162.9 $\pm$ 27.4
phishing	iP-DCA($\epsilon = 0$)	0.29 $\pm$ 0.00	0.09 $\pm$ 0.00	357.9 $\pm$ 95.2
	iP-DCA($\epsilon = 10^{-2}$)	0.29 $\pm$ 0.00	0.09 $\pm$ 0.00	222.1 $\pm$ 18.9
	iP-DCA($\epsilon = 10^{-4}$)	0.29 $\pm$ 0.00	0.09 $\pm$ 0.00	215.4 $\pm$ 46.5

The number of hyperparameters for the datasets "mushrooms" and "phishing" are $n + 1 = 112$ and 69 respectively.

Reference on application of bilevel programs in machine learning

- R.S. Liu, Y.H. Liu, S.Z. Zeng and J. Zhang, Towards gradient-based bilevel optimization with non-convex followers and beyond, NeurIPS Spotlight, 2021.
- R.S. Liu, X. Liu, X.M. Yuan, S.Z. Zeng and J. Zhang, A value-function-based interior-point method for non-convex bilevel optimization, ICML 2021.
- R.S. Liu, P.Mu, X.M. Yuan, S.Z. Zeng and J. Zhang, A generic first-order algorithmic framework for bi-level programming beyond lower-level singleton, ICML 2020.

Reference

- R.T. Rockafellar: Convex Analysis, Princeton University Press, Princeton 1970. Convex analysis theory used can be found in this book.
- H.A.L. Thi and D.T. Dinh: Advanced Computational Methods for Knowledge Engineering, DC programming and DCA for general DC programs. pp. 15-35. Springer, Cham, Switzerland (2014). Classical DCA can be found in this paper.
- JY, Xiaoming Yuan, Shangzhi Zeng and Jin Zhang 2021, Difference of Convex Algorithms for Bilevel Programs with Applications in Hyperparameter Selection, Revised for Math. Program., arXiv (2102.09006). The results reported are mainly based on this paper.

An open source Python version of our algorithm on SV bilevel model selection is provided on *github.com/SUSTech-Optimization*.

- Thank You -