Journal of Machine Learning Research 22 (2021) 1-47 Submitted 7/18; Revised 6/21; Published 9/21

On ℓ_p-hyperparameter Learning via Bilevel Nonsmooth Optimization

Takayuki Okuno
TAKAYUKI.OKUNO.KS@RIKEN.JP
Center for Advanced Intelligence Project, RIKEN
Tokyo 103-0027, Japan

Akiko Takeda
TAKEDA@MIST.I.U-TOKYO.AC.JP
Graduate School of Information Science and Technology,
The University of Tokyo
Tokyo 113-8656, Japan;
Center for Advanced Intelligence Project, RIKEN
Tokyo 103-0027, Japan

Akihiro Kawana
KAWANA.AK.PP@GMAIL.COM
Department of Industrial Engineering and Economics,
Tokyo Institute of Technology
Tokyo 152-8550, Japan
(This research was conducted when he was a student at Tokyo Institute of Technology, and is completely irrevalent to the present company.)

Motokazu Watanabe
MWATANABE@G.ECC.U-TOKYO.AC.JP
Department of Mathematical Informatics,
The University of Tokyo
Tokyo 113-8656, Japan;
Present Address: Tokio Marine & Nichido Fire Insurance Co., Ltd., Tokyo, Japan
(This research was conducted when he was a student at The University of Tokyo, and is completely irrevalent to the present company.)

Editor: Ryota Tomioka

Abstract

We propose a bilevel optimization strategy for selecting the best hyperparameter value for the nonsmooth ℓ_p regularizer with $0 < p \leq 1$. The concerned bilevel optimization problem has a nonsmooth, possibly nonconvex, ℓ_p-regularized problem as the lower-level problem. Despite the recent popularity of nonconvex ℓ_p-regularizer and the usefulness of bilevel optimization for selecting hyperparameters, algorithms for such bilevel problems have not been studied because of the difficulty of ℓ_p-regularizer.

Our contribution is the proposal of the first algorithm equipped with a theoretical guarantee for finding the best hyperparameter of ℓ_p-regularized supervised learning problems. Specifically, we propose a smoothing-type algorithm for the above mentioned bilevel optimization problems and provide a theoretical convergence guarantee for the algorithm. Indeed, since optimality conditions are not known for such bilevel optimization problems so far, new necessary optimality conditions, which are called the SB-KKT conditions, are derived and it is shown that a sequence generated by the proposed algorithm actually accumulates at a point satisfying the SB-KKT conditions under some mild assumptions. The proposed algorithm is simple and scalable as our numerical comparison to Bayesian optimization and grid search indicates.

Keywords: Hyperparameter optimization, bilevel optimization, ℓ_p-regularizer, smoothing method,

1. Introduction

Hyperparameters are parameters that are set manually outside of a learning algorithm in the context of machine learning. Hyperparameters often play important roles in exhibiting a high prediction performance. For example, a regularization parameter controls a trade-off between the regularization (that is, model complexity) and the training set error (that is, empirical error). If the hyperparameters are tuned properly, the predictive performance of learning algorithms will be increased.

Hyperparameter optimization or learning is the task of finding (near) optimal values of hyperparameters. There are mainly a few methods currently in use for supervised learning. The most popular one would be grid search. The method is to divide the space of possible hyperparameter values into regular intervals (a grid), train a learning model using training data for all values on the grid sequentially or preferably in parallel, and choose the best one with the highest prediction accuracy tested on validation, for example, by using cross validation.

There are other techniques for hyperparameter optimization; random search that evaluates learning models for randomly sampled hyperparameter values or more sophisticated method called Bayesian optimization (Mockus et al., 1978). To find a classifier/regressor with good prediction performance, it is reasonable to minimize the validation error in terms of hyperparameters. However we do not know the explicit form of the validation error function, say $\bar{f}$, represented with hyperparameter, while we are able to compute the validation error of a classifier/regressor obtained with given hyperparameters $\boldsymbol{\lambda}$, namely, $\bar{f}(\boldsymbol{\lambda})$. For such a black-box (meaning unknown) objective function $\bar{f}$, Bayesian optimization algorithms use previous observational values $\bar{f}(\boldsymbol{\lambda})$ at some hyperparameter values $\boldsymbol{\lambda}$ to determine the next point $\boldsymbol{\lambda}^+$ to evaluate $\bar{f}(\boldsymbol{\lambda}^+)$. This is based on the assumption that the function $\bar{f}$ is described by a Gaussian process as a prior. There are still essential questions unresolved; how to select a kernel for the Gaussian process, how to select the range of values to search in, and lots of implementation details. As regards a comprehensive survey of hyperparameter optimization, we refer to the article (Feurer and Hutter, 2019).

Bilevel optimization is a more direct approach for finding a best set of hyperparameter values. A bilevel optimization problem consists of two-level optimization problems; the upper-level problem minimizes the validation error in terms of hyperparameters and the lower-level problem finds a best fit line for training data combined with a regularizer using given hyperparameter values. Actually, the spirit of bilevel optimization underlies the methods introduced above. As mentioned below, some existing works pointed out the usefulness of the bilevel formulation for some classes of hyperparameters. However, there is still a lot of room to pursue the bilevel approach further, in particular, for hyperparameter optimization of nonsmooth regularizers.

1.1 Our Contribution

The purpose of this paper is to provide a bilevel optimization approach for finding a best set of hyperparameter values for the nonsmooth ℓ_p ($p \leq 1$) regularizer. The nonsmooth bilevel optimization approaches examined here are entirely novel in the field of mathematical optimization too. In recent years, research on sparse optimization using nonconvex nonsmooth regularizers has been actively conducted in machine learning (Gong et al., 2013; Hu et al., 2017), signal/image processing (Chen et al., 2010; Hintermüller and Wu, 2013; Wen et al., 2017; Marjanovic and Solo, 2013), and continuous optimization (Ge et al., 2011; Lai and Wang, 2011; Bian and Chen, 2013; Chen et al., 2014; Bian et al., 2015; Bian and Chen, 2017). In particular, for the purpose of finding a sparse solution, the ℓ_p-regularizer with $0 < p < 1$ is reported to be effective in wide applications such as matrix completion (Marjanovic and Solo, 2012), de-noising (Marjanovic and Solo, 2013), compressing sensing (Zheng et al., 2016; Wen et al., 2016; Weng et al., 2016), CT (computed tomography) reconstruction (Miao and Yu, 2016), machine learning (Xu et al., 2012) and so on. Refer to the survey article (Wen et al., 2018) concerning nonconvex regularizers including the ℓ_p-regularizer, and also see references therein. In spite of plenty of researches supporting the efficiency of the ℓ_p-regularizer, there exist fewer studies on bilevel optimization approaches to hyperparameter learning or optimization of this regularizer. A possible specific reason is that tractable optimality conditions for the arising bilevel problem have not been developed yet because of the ℓ_p-regularizer's nonsmoothness or high nonconvexity when $p < 1$. Moreover, there are no practical algorithms ensured of convergence to a meaningful point for that nonsmooth bilevel problem.

Our contribution in this paper is the proposal of an algorithm with a theoretical convergence guarantee for solving an ℓ_p-hyperparameter optimization problem, namely, the problem of finding the best hyperparameter of an ℓ_p-regularized supervised learning problem. Specifically, we first formulate it as a bilevel optimization problem with a nonsmooth and nonconvex lower-level problem having the ℓ_p-regularizer. Since no optimality conditions have been explored adequately for such bilevel problems so far, we develop new optimality conditions, named scaled bilevel KKT (SB-KKT) conditions. As a matter of fact, the SB-KKT conditions can be cast as an extension of the scaled first-order optimality conditions for some class of non-Lipschitz optimization problems originally given in the articles (Chen et al., 2010, 2013; Bian and Chen, 2017). We prove that these conditions are nothing but necessary optimality conditions for the one-level optimization problem acquired by replacing the lower-level problem with its scaled first-order optimality conditions. We moreover propose an iterative algorithm for solving the bilevel optimization problem with a nonsmooth and nonconvex lower-level problem. One natural way for tackling such a problem would be formulating it as a one-level problem by replacing its lower-level-problem constraints with the first-order optimality condition formed by the subdifferential (that is, the set of subgradients) of the ℓ_p-regularizer. However, it is still nontrivial how we solve the resulting one-level problem with point-to-set mapping constraints. To avoid this difficulty, we apply a smoothing technique for the ℓ_p-regularizer, which enables us to make use of a gradient of the smoothed regularizer. As a result, we obtain a one-level problem whose constraints are represented in terms of only smooth equations and inequalities. In the presented smoothing algorithm, we generate a sequence of KKT solutions of the smoothed problems while we control the degree of smoothing approximation. We will prove that a sequence generated by this algorithm accumulates at a point satisfying the SB-KKT conditions under some mild assumptions. Numerical experiments support the scalability of our algorithm compared to Bayesian optimization and grid search. Finally, we discuss extension of the SB-KKT conditions and the proposed algorithm to other regularizers such as SCAD and MCP.

1.2 Related Work on Bilevel Approach

Most existing bilevel optimization models assume convexity and/or smoothness for all functions or at least once differentiability for the lower-level objective functions. If it is not once differentiable, we need to overcome the difficulty of selecting a subgradient to guarantee descent of the upper-level gradient when solving such a problem.

Bilevel Formulations for Hyperparameter Opt. There are no existing works on bilevel hyperparameter optimization approach for our model and existing works are restricted to smooth and convex machine learning models. A pioneer work in the line was by Bennett et al. (2006, 2008). They formulated the selection technique of cross-validation for support vector regression as a bilevel optimization problem, equivalently transformed it into a one-level nonconvex optimization problem whose constraints are the Karush-Kuhn-Tucker (KKT) optimality conditions of the lower-level problem and proposed two approaches to solve the nonconvex problem. Moore et al. (2009, 2011) gave a bilevel optimization formulation for a nonsmooth and convex machine learning model, support vector regression (SVR), while their proposed algorithms assume that the lower-level objective functions are at least once differentiable. Pedregosa (2016) gave a bilevel optimization formulation for more general supervised learning problems, but the assumption of differentiability has been still imposed for all functions. More recently, Franceschi et al. (2018) gave a unified bilevel perspective on hyperparameter optimization and meta learning, and presented a gradient-based algorithm using automatic differentiation techniques, which assumes the smoothness of the lower-level objective function.

Bilevel Optimization Algorithms As far as we investigated, the convergence analysis for bilevel problems with nonconvex nonsmooth regularizers has not been studied before. Many studies on bilevel optimization in optimization community transform bilevel optimization problems into the one-level formulations via the first-order optimality conditions for lower-level problems by assuming the differentiability of the functions, and focus on investigating theoretical properties for constraint qualifications and optimality conditions. See, for example, (Ye and Zhu, 1995; Dempe et al., 2006; Dempe and Zemkoho, 2011, 2013; Dempe et al., 2015). Recently, Ochs et al. (2016) proposed techniques for solving bilevel optimization problems with non-smooth “convex” lower level problems. They considered a gradient-based method for the optimization problem obtained by substituting a smoothly approximated solution mapping of the lower-level problem into the upper level problem. However, theoretical analysis concerning the limiting behavior of the derivatives of the approximated solution mappings was left to future work and the proposed method was written to be heuristic in the paper. Kunisch and Pock (2013) and Rosset (2009) considered bilevel optimization problems having the ℓ_p-regularizer, which are similar to our problem, but the p was mainly restricted to 1 or 2. Especially, the case of $p = 0.5$ only appears in the numerical experiments in Kunisch and Pock (2013) without any theoretical support, though some convergence analysis is shown for the semismooth Newton algorithms for the case of $p = 1$.

Another stream of bilevel algorithms is based on the reformulation as one-level problem by replacing the lower-level problem with a dynamical system, which arises in an iterative algorithm such as proximal gradient-type methods for solving the lower-level problem. The approach is employed for hyperparameter optimization by, for example, Lorraine et al. (2020), Franceschi et al. (2017, 2018), Maclaurin et al. (2015), and Shaban et al. (2019). In their theoretical analysis, nonconvex and nonsmooth functions, which we will handle in this paper, are not supposed to be contained by the lower-level objective one.

Notations. In this paper, we often denote a vector $\boldsymbol{z} \in \mathbb{R}^d$ by $\boldsymbol{z} = (z_1, z_2, \dots, z_d)^\top$ and write $\lim_{\ell \in L \to \infty} \boldsymbol{z}^\ell = \boldsymbol{z}^*$ to represent that, given a sequence $\{\boldsymbol{z}^\ell\}$, a subsequence $\{\boldsymbol{z}^\ell\}_{\ell \in L}$ with $L \subseteq$

$\{1, 2, \dots, \}$ converges to $\boldsymbol{z}^*$. The ℓ-th vector $\boldsymbol{z}^\ell \in \mathbb{R}^d$ is often represented as $\boldsymbol{z}^\ell := (z_1^\ell, z_2^\ell, \dots, z_d^\ell)^\top$. We also denote the d-dimensional non-negative (positive) orthants by $\mathbb{R}^d_{+(++)} := \{\boldsymbol{z} \in \mathbb{R}^d \mid z_i \geq (>) 0 \ (i = 1, 2, \dots, d)\}$. For a set of vectors $\{\boldsymbol{v}_i\}_{i \in I} \subseteq \mathbb{R}^m$ with $I := \{i_1, i_2, \dots, i_p\}$, we define $(\boldsymbol{v}_i)_{i \in I} := (\boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \dots, \boldsymbol{v}_{i_p}) \in \mathbb{R}^{m \times p}$. We denote the sign function by $\mathrm{sgn} : \mathbb{R} \to \{-1, 0, +1\}$, that is, $\mathrm{sgn}(x) := 1 \ (x > 0)$, $0 \ (x = 0)$, and $-1 \ (x < 0)$ for any $x \in \mathbb{R}$.

For a differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, we denote the gradient function from $\mathbb{R}^n$ to $\mathbb{R}^n$ by ∇h, that is, $\nabla h(\boldsymbol{x}) := (\frac{\partial h(\boldsymbol{x})}{\partial x_1}, \dots, \frac{\partial h(\boldsymbol{x})}{\partial x_n})^\top \in \mathbb{R}^n$ for $\boldsymbol{x} \in \mathbb{R}^n$, where $\frac{\partial h(\boldsymbol{x})}{\partial x_i}$ stands for the partial differential of h with respect to x_i for $i = 1, 2, \dots, n$. To express the gradient of h with respect to a sub-vector $\tilde{\boldsymbol{x}} := (x_i)_{i \in I}^\top$ of $\boldsymbol{x}$ with $I := \{i_1, i_2, \dots, i_p\} \subseteq \{1, 2, \dots, n\}$, we write $\nabla_{\tilde{\boldsymbol{x}}} h(\boldsymbol{x}) := \left(\frac{\partial h(\boldsymbol{x})}{\partial x_{i_1}}, \frac{\partial h(\boldsymbol{x})}{\partial x_{i_2}}, \dots, \frac{\partial h(\boldsymbol{x})}{\partial x_{i_p}}\right)^\top \in \mathbb{R}^{|I|}$. We often write $\nabla g(\boldsymbol{x})|_{\boldsymbol{x} = \bar{\boldsymbol{x}}}$ $(\nabla_{\tilde{\boldsymbol{x}}} g(\boldsymbol{x})|_{\boldsymbol{x} = \bar{\boldsymbol{x}}})$ or $\nabla h(\bar{\boldsymbol{x}})$ $(\nabla_{\tilde{\boldsymbol{x}}} h(\bar{\boldsymbol{x}}))$ to represent the (partial) gradient value of g at $\boldsymbol{x} = \bar{\boldsymbol{x}}$. Moreover, when h is twice differentiable, we denote the Hessian of h by $\nabla^2 h : \mathbb{R}^n \to \mathbb{R}^{n \times n}$, that is, $\nabla^2 h(\boldsymbol{x}) := \left(\frac{\partial^2 h(\boldsymbol{x})}{\partial x_i \partial x_j}\right)_{1 \leq i, j \leq n} \in \mathbb{R}^{n \times n}$.

Organization of the Paper The rest of this paper is organized as follows: In Section 2, we describe our problem setting precisely. In Section 3, we propose a smoothing algorithm for solving the targeted problem. In Section 4, we present new necessary optimality conditions of the problem, already refereed to as the SB-KKT conditions. We also conduct the convergence analysis of the proposed smoothing algorithm. In Section 5, we examine the efficiency of the proposed algorithm by means of numerical experiments using real data sets. In Section 6, we discuss extension of the proposed algorithm to other classes of problems. Finally, in Section 7, we conclude this paper. In Appendix, we provide some proofs omitted in the main part together with other supplementary materials.

2. Formulation

We consider the following bilevel optimization problem with a nonsmooth, possibly nonconvex, lower-level problem:

$$\min_{\boldsymbol{w}^*_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}} \ f(\boldsymbol{w}^*_{\boldsymbol{\lambda}}) \ \text{ s.t. } \ \boldsymbol{w}^*_{\boldsymbol{\lambda}} \in \operatorname*{argmin}_{\boldsymbol{w} \in \mathbb{R}^n} \left(g(\boldsymbol{w}) + \sum_{i=1}^r \lambda_i R_i(\boldsymbol{w}) \right), \ \boldsymbol{\lambda} \geq \boldsymbol{0}. \tag{1}$$

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is once continuously differentiable, $\boldsymbol{\lambda} := (\lambda_1, \lambda_2, \dots, \lambda_r)^\top \in \mathbb{R}^r$, $R_1(\boldsymbol{w}) := \|\boldsymbol{w}\|_p^p = \sum_{i=1}^n |w_i|^p$ $(0 < p \leq 1)$, and the functions $R_2, \cdots, R_r$, and g are twice continuously differentiable functions. We call the whole problem (1) and $\min_{\boldsymbol{w} \in \mathbb{R}^n} g(\boldsymbol{w}) + \sum_{i=1}^r \lambda_i R_i(\boldsymbol{w})$ the upper- and lower-level problem, respectively. To make our notation simple, we often use the function

$$G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) := g(\boldsymbol{w}) + \sum_{i=2}^r \lambda_i R_i(\boldsymbol{w}),$$

with $\bar{\boldsymbol{\lambda}} := (\lambda_2, \dots, \lambda_r)^\top \in \mathbb{R}^{r-1}$ for expressing the lower-level problem as

$$\min_{\boldsymbol{w} \in \mathbb{R}^n} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 R_1(\boldsymbol{w}).$$

Note that the function R_1 is nonconvex when $p < 1$ and nonsmooth, though some differentiability is assumed for other terms. We also remark that the proposed smoothing algorithm can be tailored to problems that contain multiple nonsmooth regularizers as long as suitable smoothing functions (see Section 3 for the definition) are found. Nonetheless, it is unclear whether the theoretical analysis can be established under such a setting.

2.1 Examples of Functions g, $\sum_{i=1}^{r} \lambda_i R_i$, and f

When using the following loss function as the function g:

- $g(\boldsymbol{w}) = \sum_{i=1}^{m_{tr}} (\tilde{y}_i - \tilde{\boldsymbol{x}}_i^\top \boldsymbol{w})^2$ for training samples $(\tilde{y}_i, \tilde{\boldsymbol{x}}_i) \in \mathbb{R} \times \mathbb{R}^n, i = 1, \cdots, m_{tr}$
- $g(\boldsymbol{w}) = \sum_{i=1}^{m_{tr}} \log(1 + \exp(-\tilde{y}_i \tilde{\boldsymbol{x}}_i^\top \boldsymbol{w}))$ for training samples $(\tilde{y}_i, \tilde{\boldsymbol{x}}_i) \in \{+1, -1\} \times \mathbb{R}^n, i = 1, \cdots, m_{tr}$,

the lower-level optimization problem in (1) corresponds to minimizing the ℓ_2-loss function for regression and the logistic-loss function for binary classification, respectively, combined with some regularization including $\|\boldsymbol{w}\|_p^p$ for a given hyperparameter vector $\boldsymbol{\lambda}$. This type of problem whose regularizer includes $\|\boldsymbol{w}\|_p^p$ is called a sparse optimization problem. Various well-known sparse regularizers can be expressed by $\sum_{i=1}^{r} \lambda_i R_i(\boldsymbol{w})$. For example,

- ⋆ ℓ_1 regularizer: $\lambda_1 \|\boldsymbol{w}\|_1$,
- ⋆ elastic net regularizer: $\lambda_1 \|\boldsymbol{w}\|_1 + \lambda_2 \|\boldsymbol{w}\|_2^2$,
- ⋆ nonconvex regularizer: $\lambda_1 \|\boldsymbol{w}\|_q^q$ with $0 < q < 1$.

What we want to do is to find the best hyperparameter values of $\boldsymbol{\lambda}$ which lead to small validation error. The upper-level problem can find such values for $\boldsymbol{\lambda}$. By setting the same loss function with g for f but defined by validation samples $(\hat{y}_j, \hat{\boldsymbol{x}}_j), j = 1, \cdots, m_{\text{val}}$, the upper-level problem finds the best hyperparameter values which minimize the validation error, which is defined by $f(\boldsymbol{w}) = \sum_{i=1}^{m_{\text{val}}} (\hat{y}_i - \hat{\boldsymbol{x}}_i^\top \boldsymbol{w})^2$ for the ℓ_2-loss or $f(\boldsymbol{w}) = \sum_{i=1}^{m_{\text{val}}} \log(1 + \exp(-\hat{y}_i \hat{\boldsymbol{x}}_i^\top \boldsymbol{w}))$ for the logistic-loss.

3. Smoothing Method for Nonconvex Nonsmooth Bilevel Program

For problem (1), one may think of the one-level problem obtained by replacing the lower problem constraint with its first-order optimality condition (Rockafellar and Wets, 2009, 10.1 Theorem) represented in terms of (general) subgradient[1], namely,

$$\min_{\boldsymbol{w}, \boldsymbol{\lambda}} \; f(\boldsymbol{w}) \;\; \text{s.t.} \;\; \boldsymbol{0} \in \partial_{\boldsymbol{w}}(G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 R_1(\boldsymbol{w})), \;\; \boldsymbol{\lambda} \geq \boldsymbol{0}. \tag{2}$$

Notice that $G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 R_1(\boldsymbol{w})$ is not convex with respect to $\boldsymbol{w}$ generally. Hence, the feasible region of (2) can be larger than that of the original problem (1) because not only the global optimal solutions of the lower-level problem but also its local optimal solutions are feasible solutions for (2). In that sense, problem (2) is modified from the original one, but solving it may lead to better prediction performance because the best hyperparameter λ is searched in the wider space and above all, there is no way to solve the bilevel optimization problem (1) as it is.

[1] For precise definitions of a subgradient of a nonconvex function, see Appendix A.2 in this paper or Chapter 8 of the book (Rockafellar and Wets, 2009).

3.1 Smoothing method

In our approach for tackling problem (1), we will use the smoothing method, which is one of the most powerful methodologies developed for solving nonsmooth equations, nonsmooth optimization problems, and so on. Fundamentally, the smoothing method solves smoothed optimization problems or equations sequentially to produce a sequence converging to a point that satisfies some optimality conditions of the original nonsmooth problem. The smoothed problems solved therein are obtained by replacing the nonsmooth functions with so-called smoothing functions.

Let $\varphi_0 : \mathbb{R}^n \to \mathbb{R}$ be a nonsmooth function. Then, we say that $\varphi : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ is a smoothing function of φ_0 when (i) $\varphi(\cdot, \cdot)$ is continuous and $\varphi(\cdot, \mu)$ is continuously differentiable for any $\mu > 0$; (ii) $\lim_{\tilde{\boldsymbol{w}} \to \boldsymbol{w}, \mu \to 0+} \varphi(\tilde{\boldsymbol{w}}, \mu) = \varphi_0(\boldsymbol{w})$ for any $\boldsymbol{w} \in \mathbb{R}^n$. In particular, we call $\mu \geq 0$ a smoothing parameter. For more details on smoothing methods, see the comprehensive survey article (Chen, 2012) and also relevant articles (Nesterov, 2005; Beck and Teboulle, 2012).

3.2 Our approach

We propose a smoothing based method for solving (1). In the method, we replace the nonsmooth, possibly nonconvex, term $R_1(\boldsymbol{w}) = \|\boldsymbol{w}\|_p^p$ in (1) by the following smoothing function:

$$\varphi_\mu(\boldsymbol{w}) := \sum_{i=1}^{n} (w_i^2 + \mu^2)^{\frac{p}{2}}.$$

We then have the following bilevel problem approximating the original one (1):

$$\min_{\boldsymbol{w}^*_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}} \; f(\boldsymbol{w}^*_{\boldsymbol{\lambda}}) \;\; \text{s.t.} \;\; \boldsymbol{w}^*_{\boldsymbol{\lambda}} \in \operatorname*{argmin}_{\boldsymbol{w} \in \mathbb{R}^n} \left(G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 \varphi_\mu(\boldsymbol{w}) \right), \; \boldsymbol{\lambda} \geq \boldsymbol{0}$$

which naturally leads to the following one-level problem:

$$\begin{array}{cc} \min\limits_{\boldsymbol{w}, \boldsymbol{\lambda}} & f(\boldsymbol{w}) \\ \text{s.t.} & \nabla_{\boldsymbol{w}} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 \nabla \varphi_\mu(\boldsymbol{w}) = \boldsymbol{0} \\ & \boldsymbol{\lambda} \geq \boldsymbol{0}. \end{array} \tag{3}$$

Note that problem (3) is smooth since the function φ_μ is twice continuously differentiable [2] when $\mu \neq 0$. Hence, we can consider the Karush-Kuhn-Tucker (KKT) conditions for this problem.

Let us explain the proposed method in detail. To this end, for a parameter $\hat{\varepsilon} > 0$, we define an $\hat{\varepsilon}$-approximate KKT point for problem (3). We say that $(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\zeta}, \boldsymbol{\eta}) \in \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^n \times \mathbb{R}^r$ is an $\hat{\varepsilon}$-approximate KKT point for (3) if there exists a vector $(\boldsymbol{\varepsilon}_1, \varepsilon_2, \boldsymbol{\varepsilon}_3, \boldsymbol{\varepsilon}_4, \varepsilon_5) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^{r-1} \times \mathbb{R}^n \times \mathbb{R}$ such that

$$\nabla f(\boldsymbol{w}) + \left(\nabla^2_{\boldsymbol{w}\boldsymbol{w}} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 \nabla^2 \varphi_\mu(\boldsymbol{w}) \right) \boldsymbol{\zeta} = \boldsymbol{\varepsilon}_1, \tag{4}$$

$$\nabla \varphi_\mu(\boldsymbol{w})^\top \boldsymbol{\zeta} - \eta_1 = \varepsilon_2, \tag{5}$$

$$\nabla R_i(\boldsymbol{w})^\top \boldsymbol{\zeta} - \eta_i = (\boldsymbol{\varepsilon}_3)_i \;\; (i = 2, 3, \ldots, r), \tag{6}$$

$$\nabla_{\boldsymbol{w}} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 \nabla \varphi_\mu(\boldsymbol{w}) = \boldsymbol{\varepsilon}_4, \tag{7}$$

$$\boldsymbol{0} \leq \boldsymbol{\lambda}, \; \boldsymbol{0} \leq \boldsymbol{\eta}, \; \boldsymbol{\lambda}^\top \boldsymbol{\eta} = \varepsilon_5, \tag{8}$$

[2] Huber's function (Beck and Teboulle, 2012) is a popular smoothing function of $R_1(\cdot)$, but is not twice continuously differentiable.

Algorithm 1 Smoothing Method for Nonsmooth Bilevel Program

```
Require: Choose $\mu_0 \neq 0$, $\beta_1, \beta_2 \in (0,1)$ and $\hat{\varepsilon}_0 \geq 0$. Set $k \leftarrow 0$.
repeat
    Find an $\hat{\varepsilon}_k$-approximate KKT point $(\boldsymbol{w}^{k+1}, \boldsymbol{\lambda}^{k+1}, \boldsymbol{\zeta}^{k+1}, \boldsymbol{\eta}^{k+1})$ for problem (3) with $\mu = \mu_k$.
    Update the smoothing and error parameters by $\mu_{k+1} \leftarrow \beta_1 \mu_k$ and $\hat{\varepsilon}_{k+1} \leftarrow \beta_2 \hat{\varepsilon}_k$.
    $k \leftarrow k+1$.
until convergence of $(\boldsymbol{w}^k, \boldsymbol{\lambda}^k, \boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)$.
```

and

$$\|(\boldsymbol{\varepsilon}_1, \varepsilon_2, \boldsymbol{\varepsilon}_3, \boldsymbol{\varepsilon}_4, \varepsilon_5)\| \leq \hat{\varepsilon},$$

where $\nabla^2_{\boldsymbol{w}\boldsymbol{w}} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})$ is the Hessian of G with respect to $\boldsymbol{w}$. Notice that an $\hat{\varepsilon}$-approximate KKT point is nothing but a KKT point[3] for problem (3) if $\hat{\varepsilon} = 0$. Hence, $\boldsymbol{\zeta} \in \mathbb{R}^n$ and $\boldsymbol{\eta} \in \mathbb{R}^r$ are regarded as approximate Lagrange multiplier vectors corresponding to the equality constraint $\nabla_{\boldsymbol{w}} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 \nabla \varphi_\mu(\boldsymbol{w}) = \boldsymbol{0}$ and the inequality constraints $\boldsymbol{\lambda} \geq \boldsymbol{0}$, respectively. The proposed algorithm produces a sequence of $\hat{\varepsilon}$-approximate KKT points for problem (3) while decreasing the values of $\hat{\varepsilon}$ and μ to 0. Precisely, it is described as in Algorithm 1.

Though, in Algorithm 1, we do not designate any means for computing an $\hat{\varepsilon}$-approximate KKT point for (3), sequential quadratic programming (SQP) methods (Nocedal and Wright, 2006) are promising candidates. However, since such SQP methods are designed for solving general constrained problems, we may develop more efficient algorithms by exploiting structure of individual problems. This issue will be discussed later in Section 5. See also Appendix B. As for practical stopping criteria of Algorithm 1, we make use of the scaled bilevel SB-KKT conditions studied in the subsequent section.

4. Theoretical Results

In this section, we will prove the global convergence of Algorithm 1 by investigating an accumulation point of a sequence generated by that algorithm. For this purpose, in Section 4.1, we first present new optimality conditions for the original bilevel problem (1), named *scaled bilevel KKT (SB-KKT)* conditions. Moreover, in Section 4.2, we prove that any accumulation point of a sequence generated by Algorithm 1 actually satisfies the SB-KKT conditions under some assumptions.

Throughout the section, we often use the following notations for $\boldsymbol{w} \in \mathbb{R}^n$:

$$I(\boldsymbol{w}) := \{i \in \{1, 2, \ldots, n\} \mid w_i = 0\}, \ |\boldsymbol{w}|^p := (|w_1|^p, |w_2|^p, \ldots, |w_n|^p)^\top .$$

4.1 SB-KKT Conditions

Now, we define the SB-KKT conditions for problem (1):

Definition 1 *We say that the scaled bilevel Karush-Kuhn-Tucker (SB-KKT) conditions hold at* $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*) \in \mathbb{R}^n \times \mathbb{R}^r$ *for problem* (1) *when there exists a pair of vectors* $(\boldsymbol{\zeta}^*, \boldsymbol{\eta}^*) \in \mathbb{R}^n \times \mathbb{R}^r$ *such*

[3] Note that (5) and (6) with $(\varepsilon_2, (\boldsymbol{\varepsilon}_3)_2, \ldots, (\boldsymbol{\varepsilon}_3)_r) = \boldsymbol{0}$ can be obtained from

$$\nabla_{\boldsymbol{\lambda}} f(\boldsymbol{w}) + \nabla_{\boldsymbol{\lambda}} \left(\left(\nabla_{\boldsymbol{w}} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 \nabla \varphi_\mu(\boldsymbol{w}) \right)^\top \boldsymbol{\zeta} \right) - \boldsymbol{\eta} = \boldsymbol{0}.$$

that

$$
\begin{aligned}
\boldsymbol{W}_*^2 \nabla f(\boldsymbol{w}^*) + \boldsymbol{H}(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)\zeta^* = \boldsymbol{0}, \quad (9)\\
\boldsymbol{W}_* \nabla_{\boldsymbol{w}} G(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*) + p\lambda_1^* |\boldsymbol{w}^*|^p = \boldsymbol{0}, \quad (10)\\
p \textstyle\sum_{i \notin I(\boldsymbol{w}^*)} \operatorname{sgn}(w_i^*) |w_i^*|^{p-1} \zeta_i^* = \eta_1^*, \quad (11)\\
\zeta_i^* = 0 \ (i \in I(\boldsymbol{w}^*)), \quad (12)\\
\nabla R_i(\boldsymbol{w}^*)^\top \boldsymbol{\zeta}^* = \eta_i^* \ \ (i = 2, 3, \ldots, r), \quad (13)\\
\boldsymbol{0} \le \boldsymbol{\lambda}^*, \ \boldsymbol{0} \le \boldsymbol{\eta}^*, \ (\boldsymbol{\lambda}^*)^\top \boldsymbol{\eta}^* = 0, \quad (14)
\end{aligned}
$$

where $\boldsymbol{W}_* := \operatorname{diag}(\boldsymbol{w}^*)$. *Here, we write*

$$\boldsymbol{H}(\boldsymbol{w}, \boldsymbol{\lambda}) := \boldsymbol{W}^2 \nabla_{\boldsymbol{w}}^2 G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 p(p-1)\operatorname{diag}(|\boldsymbol{w}|^p)$$

with $\boldsymbol{W} := \operatorname{diag}(\boldsymbol{w})$ *for* $\boldsymbol{w} \in \mathbb{R}^n$ *and* $\boldsymbol{\lambda} \in \mathbb{R}^r$. *In particular, we call a point* $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*) \in \mathbb{R}^n \times \mathbb{R}^r$ *satisfying the above conditions* (9)–(14) *an SB-KKT point for problem* (1).

In fact, $(\boldsymbol{0}, \boldsymbol{\lambda}^*)$ is a trivial SB-KKT point for any $\boldsymbol{\lambda}^*$. This can be checked by setting $(\boldsymbol{\zeta}^*, \boldsymbol{\eta}^*) = (\boldsymbol{0}, \boldsymbol{0})$ in the conditions (9)–(14). Experimentally, started from a point apart from such a trivial point, Algorithm 1 finds non-trivial SB-KKT points in many cases.

We next prove that the SB-KKT conditions are necessary optimality conditions for a certain one-level problem. For this purpose, we derive the *scaled first-order necessary condition* (Chen et al., 2010) for the lower-level problem in (1):

$$\min_{\boldsymbol{w} \in \mathbb{R}^n} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 \|\boldsymbol{w}\|_p^p. \quad (15)$$

We say that the scaled first-order necessary condition of (15) holds at $\boldsymbol{w}^*$ if

$$\boldsymbol{W}_* \nabla_{\boldsymbol{w}} G(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}) + p\lambda_1 |\boldsymbol{w}^*|^p = \boldsymbol{0}. \quad (16)$$

Indeed, a local optimum $\boldsymbol{w}^*$ of (15) satisfies the above condition. This fact can be verified easily by following the proof of Theorem 2.1 in the article (Chen et al., 2010). The above scaled condition was originally presented in the articles (Chen et al., 2010, 2013; Bian and Chen, 2017) for some optimization problems admitting non-Lipschitz functions.

As in deriving (2), we obtain the following one-level problem by replacing the lower problem in (1) with the scaled first-order necessary condition (16):

$$\min_{\boldsymbol{w}, \boldsymbol{\lambda}} \ f(\boldsymbol{w}) \ \text{ s.t. } \ \boldsymbol{W} \nabla_{\boldsymbol{w}} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + p\lambda_1 |\boldsymbol{w}|^p = \boldsymbol{0}, \ \boldsymbol{\lambda} \ge \boldsymbol{0}. \quad (17)$$

As well as (2), the feasible region of (17) includes not only the global optimal solution of the lower-level problem in the original problem (1) but also its local solutions. Notice that the above problem is still nonsmooth due to the existence of $|\boldsymbol{w}|^p$.

The following theorem states that the SB-KKT conditions are necessary optimality conditions for (17). Here, we just give an outline of the proof and defer its detail to Appendix A.1.

Theorem 2 *Let* $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*) \in \mathbb{R}^n \times \mathbb{R}^r$ *be a local optimum of* (17). *Then,* $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ *together with some vectors* $\boldsymbol{\zeta}^* \in \mathbb{R}^n$ *and* $\boldsymbol{\eta}^* \in \mathbb{R}^r$ *satisfies the SB-KKT conditions* (9)–(14) *under an appropriate constraint qualification concerning the constraints* $\frac{\partial G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})}{\partial w_i} + p \operatorname{sgn}(w_i) \lambda_1 |w_i|^{p-1} = 0 \ (i \notin I(\boldsymbol{w}^*)), \ w_i = 0 \ (i \in I(\boldsymbol{w}^*))$, *and* $\boldsymbol{\lambda} \ge \boldsymbol{0}$.

Sketch of the proof: Notice that $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ is a local optimum of the following problem:

$$\begin{array}{ll} \min\limits_{\boldsymbol{w},\boldsymbol{\lambda}} & f(\boldsymbol{w}) \\ \text{s.t.} & \dfrac{\partial G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})}{\partial w_i} + p\,\mathrm{sgn}(w_i)\lambda_1 |w_i|^{p-1} = 0 \ (i \notin I(\boldsymbol{w}^*)) \\ & w_i = 0 \ (i \in I(\boldsymbol{w}^*)) \\ & \boldsymbol{\lambda} \geq \mathbf{0}. \end{array} \tag{18}$$

This is because $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ is also feasible to (18) and the feasible region of (17) is larger than that of (18). Hence, the KKT conditions hold at $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ for (18) in the presence of a constraint qualification. Finally, these KKT conditions can be equivalently transformed into the desired SB-KKT conditions. ■

In the next section, we will study convergence analysis of Algorithm 1 to an SB-KKT point. Before proceeding to the convergence analysis, let us see the relationship between the two one-level problems (2) and (17). The following lemma concerns the feasible regions of (2) and (17).

Lemma 3 *For $\boldsymbol{w} \in \mathbb{R}^n$ and $\boldsymbol{\lambda} \in \mathbb{R}^r_+$, if $\mathbf{0} \in \partial_{\boldsymbol{w}}(G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 R_1(\boldsymbol{w}))$, then $\boldsymbol{W}\nabla_{\boldsymbol{w}} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + p\lambda_1 |\boldsymbol{w}|^p = \mathbf{0}$. In particular, when $p < 1$, the converse is also true.*

Proof See Appendix A.2. ■

In view of the above lemma, we find that the feasible region of (17) is larger than that of (2) in general. However, for the case of $p < 1$, we also see that these two regions are identical. These relationships are summarized as in the following diagram.

$$\text{Feasible region of (1)} \subseteq \text{Feasible region of (2)} \left\{ \begin{array}{c} \subseteq_{p=1} \\ =_{p<1} \end{array} \right\} \text{Feasible region of (17)}$$

From this observation and Theorem 2, we can derive the following theorem immediately:

Theorem 4 *Let $p < 1$ and $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*) \in \mathbb{R}^n \times \mathbb{R}^r$ be a local optimum of* (2)*. Then, $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ together with some vectors $\boldsymbol{\zeta}^* \in \mathbb{R}^n$ and $\boldsymbol{\eta}^* \in \mathbb{R}^r$ satisfies the SB-KKT conditions* (9)–(14) *under an appropriate constraint qualification concerning the constraints $\frac{\partial G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})}{\partial w_i} + p\,\mathrm{sgn}(w_i)\lambda_1 |w_i|^{p-1} = 0 \ (i \notin I(\boldsymbol{w}^*)),\ w_i = 0 \ (i \in I(\boldsymbol{w}^*))$, and $\boldsymbol{\lambda} \geq \mathbf{0}$.*

4.2 Convergence of Algorithm 1 to an SB-KKT Point

Hereafter, for convenience of explanation, we suppose that an $\hat{\varepsilon}_{k-1}$-approximate KKT point $(\boldsymbol{w}^k, \boldsymbol{\lambda}^k, \boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)$ is a solution satisfying conditions (4)-(8) with

$$(\boldsymbol{\varepsilon}_1, \varepsilon_2, \boldsymbol{\varepsilon}_3, \boldsymbol{\varepsilon}_4, \varepsilon_5) = (\boldsymbol{\varepsilon}_1^{k-1}, \varepsilon_2^{k-1}, \boldsymbol{\varepsilon}_3^{k-1}, \boldsymbol{\varepsilon}_4^{k-1}, \varepsilon_5^{k-1}),$$

where $\{(\boldsymbol{\varepsilon}_1^k, \varepsilon_2^k, \boldsymbol{\varepsilon}_3^k, \boldsymbol{\varepsilon}_4^k, \varepsilon_5^k)\}$ is a sequence that converges to zero as $k \to \infty$.

Moreover, we suppose that the algorithm is well-defined in the sense that an $\hat{\varepsilon}_k$-approximate KKT point of (3) is found in Step 2 at every iteration, and it generates an infinite number of iteration points. In addition, we make the following assumptions:

Assumption A: Let $\{(\boldsymbol{w}^k, \boldsymbol{\lambda}^k, \boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)\} \subseteq \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^n \times \mathbb{R}^r$ be a sequence produced by the proposed algorithm. Then, the following properties hold:

A1: $\liminf_{k\to\infty} \lambda_1^k > 0$.

A2: The sequence $\{(\boldsymbol{w}^k, \boldsymbol{\lambda}^k, \boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)\}$ is bounded.

A3: Let $p = 1$ and $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ be an arbitrary accumulation point of the sequence $\{(\boldsymbol{w}^k, \boldsymbol{\lambda}^k)\}$. It then holds that $\lambda_1^* \neq \left|\frac{\partial G(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*)}{\partial w_i}\right|$ for any $i \in I(\boldsymbol{w}^*)$.

Assumption A1 means that the ℓ_p-regularization term, that is, the function R_1 works effectively. We will discuss Assumption A2 at the end of this section. Specifically, we will prove that under certain conditions, the Lagrange multiplier part $\{\boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)\}$ is actually bounded. Assumption A3 is a technical assumption for the case of $p = 1$. It indicates that, for all $i \in I(\boldsymbol{w}^*)$, zero is not situated on the boundary of the subdifferential of $G(\boldsymbol{w}, \boldsymbol{\lambda}) + \lambda\|\boldsymbol{w}\|_1$ w.r.t. w_i. Interestingly, for the case of $p < 1$, we can establish the convergence of Algorithm 1 in the absence of A3.

Under the presence of these assumptions, our goal is to prove the following convergence theorem, which motivates us to make a stopping criterion of the algorithm based on the SB-KKT conditions in the numerical experiments we will conduct later.

Theorem 5 *Suppose that Assumptions A1–A3 hold. Then, any accumulation point of* $\{(\boldsymbol{w}^k, \boldsymbol{\lambda}^k, \boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)\}$ *generated by Algorithm 1 satisfies the SB-KKT conditions* (9)–(14) *for problem* (1).

We will prove this theorem by showing that passing the approximate KKT conditions (4)-(8) to the limit yields the SB-KKT conditions. For this purpose, in particular, we have to examine how $\nabla\varphi_{\mu_k}(\boldsymbol{w}^k)$ and $\nabla^2\varphi_{\mu_k}(\boldsymbol{w}^k)$ behave in the limit. We remark that each component of $\nabla\varphi_\mu(\boldsymbol{w})$ and each diagonal one of $\nabla^2\varphi_\mu(\boldsymbol{w})$ are expressed as

$$(\nabla\varphi_\mu(\boldsymbol{w}))_i = pw_i(w_i^2 + \mu^2)^{\frac{p}{2}-1}, \tag{19}$$

$$(\nabla^2\varphi_\mu(\boldsymbol{w}))_{ii} = p(w_i^2 + \mu^2)^{\frac{p}{2}-1} + p(p-2)w_i^2(w_i^2 + \mu^2)^{\frac{p}{2}-2} \tag{20}$$

for $i = 1, 2, \ldots, n$, $\mu > 0$, and $\boldsymbol{w} \in \mathbb{R}^n$. Note that all the off-diagonal components of $\nabla^2\varphi_\mu(\boldsymbol{w})$ are zeros. We present the following proposition, whose proof is given in Appendix A.3.

Proposition 6 *Let* $\boldsymbol{w}^*$ *be the point defined in* **A3**. *Then, we have*

$$\lim_{k\to\infty} \boldsymbol{W}_k \nabla\varphi_{\mu_{k-1}}(\boldsymbol{w}^k) = p|\boldsymbol{w}^*|^p, \tag{21}$$

$$\lim_{k\to\infty} \boldsymbol{W}_k^2 \nabla^2\varphi_{\mu_{k-1}}(\boldsymbol{w}^k) = p(p-1)\mathrm{diag}(|\boldsymbol{w}^*|^p), \tag{22}$$

where $\boldsymbol{W}_k := \mathrm{diag}(\boldsymbol{w}_k)$ *for each* k.

Next, we prove that, for $i \in I(\boldsymbol{w}^*)$, the i-th diagonal component of $(\nabla^2\varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_{ii}$ diverges. The key of its proof is the approach speed of w_i ($i \in I(\boldsymbol{w}^*)$) towards zeros compared with that of the smoothing parameter μ_{k-1}. Actually, according to the next lemma, μ_{k-1} gradually approaches 0 with the speed not faster than $\max_{i\in I(\boldsymbol{w}^*)} |w_i^k|^{\frac{1}{2-p}}$. The proof will be given in Appendix A.4. Remarkably, when $p < 1$, it holds true in the absence of Assumption **A3**.

Lemma 7 *Suppose that Assumptions A1–A3 hold. Let* $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ *be an arbitrary accumulation point of* $\{(\boldsymbol{w}^k, \boldsymbol{\lambda}^k)\}$ *and* $\{(\boldsymbol{w}^k, \boldsymbol{\lambda}^k)\}_{k\in K}(\subseteq \{(\boldsymbol{w}^k, \boldsymbol{\lambda}^k)\})$ *be an arbitrary subsequence converging to* $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$. *Then, there exists some* $\gamma > 0$ *such that*

$$\mu_{k-1}^2 \geq \gamma|w_i^k|^{\frac{2}{2-p}} \ \ (i \in I(\boldsymbol{w}^*))$$

for all $k \in K$ *sufficiently large.*

From the above lemma, we can derive the following proposition.

Proposition 8 *Suppose that Assumptions A1–A3 hold. Let $\boldsymbol{w}^*$ be an arbitrary accumulation point of the sequence $\{\boldsymbol{w}^k\}$ and $\{\boldsymbol{w}^k\}_{k\in K}(\subseteq \{\boldsymbol{w}^k\})$ be an arbitrary subsequence converging to $\boldsymbol{w}^*$. Then, for any $i \in I(\boldsymbol{w}^*)$,*

$$\lim_{k\in K\to\infty}\left|(\nabla^2\varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_{ii}\right| = \infty.$$

Proof Choose $i \in I(\boldsymbol{w}^*)$ arbitrarily. Note that

$$\lim_{k\in K\to\infty}|w_i^k| = w_i^* = 0. \tag{23}$$

By Lemma 7, there is some $\gamma > 0$ such that

$$\mu_{k-1}^2 \geq \gamma|w_i^k|^{\frac{2}{2-p}} \tag{24}$$

for all $k \in K$ sufficiently large. In view of this fact, we have, for all $k \in K$ large enough,

$$\begin{aligned}\mu_{k-1}^2 + (p-1)(w_i^k)^2 &\geq \gamma|w_i^k|^{\frac{2}{2-p}} + (p-1)|w_i^k|^2\\ &= |w_i^k|^{\frac{2}{2-p}}(\gamma + (p-1)|w_i^k|^{2-\frac{2}{2-p}})\\ &\geq 0,\end{aligned} \tag{25}$$

where the second inequality can be verified by noting that $\gamma + (p-1)|w_i^k|^{2-\frac{2}{2-p}} > 0$ holds for all $k \in K$ sufficiently large because $\gamma > 0$ and $(p-1)\lim_{k\in K\to\infty}|w_i^k|^{2-\frac{2}{2-p}} = (p-1)|w_i^*|^{2-\frac{2}{2-p}} = 0$. Furthermore, notice that

$$|w_i^k|^{\frac{2}{2-p}} \geq |w_i^k|^2$$

holds for all $k \in K$ sufficiently large because $1 < \frac{2}{2-p} \leq 2$ and $|w_i^k| < 1$ for all $k \in K$ large enough by (23). Relation (24) then implies

$$\mu_{k-1}^2 \geq \gamma|w_i^k|^2. \tag{26}$$

From expression (20), it follows that

$$\begin{aligned}\left|(\nabla^2\varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_{ii}\right| &= p\left|((w_i^k)^2+\mu_{k-1}^2)^{\frac{p}{2}-2}\left((w_i^k)^2+\mu_{k-1}^2+(p-2)(w_i^k)^2\right)\right|\\ &= p\left|((w_i^k)^2+\mu_{k-1}^2)^{\frac{p}{2}-2}(\mu_{k-1}^2+(p-1)(w_i^k)^2)\right|\\ &= p((w_i^k)^2+\mu_{k-1}^2)^{\frac{p}{2}-2}\left(\mu_{k-1}^2+(p-1)(w_i^k)^2\right)\\ &\geq p\left(1+\frac{1}{\gamma}\right)^{\frac{p}{2}-2}\mu_{k-1}^{p-4}\left(\mu_{k-1}^2+(p-1)(w_i^k)^2\right)\\ &= p\left(1+\frac{1}{\gamma}\right)^{\frac{p}{2}-2}\mu_{k-1}^{p-2}\left(1+(p-1)\frac{(w_i^k)^2}{\mu_{k-1}^2}\right),\end{aligned} \tag{27}$$

where the third equality follows from (25) and the first inequality comes from (26) and $\frac{p}{2}-2 < 0$. Moreover, by (24), we see $\mu_{k-1}^{2(2-p)}/\gamma^{2-p} \geq (w_i^k)^2$ and thus have

$$\frac{\mu_{k-1}^{2-2p}}{\gamma^{2-p}} \geq \frac{(w_i^k)^2}{\mu_{k-1}^2}.$$

By this inequality, it holds that

$$\lim_{k \in K \to \infty} \left| \frac{(w_i^k)^2}{\mu_{k-1}^2} \right| \begin{cases} = 0 & (p < 1) \\ \leq \frac{1}{\gamma} & (p = 1). \end{cases}$$

Thus, we obtain

$$\lim_{k \in K \to \infty} (p-1) \left| \frac{(w_i^k)^2}{\mu_{k-1}^2} \right| = 0,$$

which together with (27) and $\lim_{k \to \infty} \mu_{k-1}^{p-2} = \infty$ implies

$$\lim_{k \in K \to \infty} \left| (\nabla^2 \varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_{ii} \right| = \infty.$$

Since $i \in I(\boldsymbol{w}^*)$ was arbitrarily chosen, the proof is complete. ■

Now, we are ready to prove Theorem 5 using Propositions 6 and 8.

Proof of Theorem 5

Consider the $\hat{\varepsilon}_{k-1}$-approximate KKT conditions (4), (6), and (7) with $(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\zeta}, \boldsymbol{\eta}) = (\boldsymbol{w}^k, \boldsymbol{\lambda}^k, \boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)$ and $(\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_3, \varepsilon_5) = (\boldsymbol{\varepsilon}_1^{k-1}, \boldsymbol{\varepsilon}_3^{k-1}, \varepsilon_5^{k-1})$. Let $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\zeta}^*, \boldsymbol{\eta}^*)$ be an arbitrary accumulation point of $\{(\boldsymbol{w}^k, \boldsymbol{\lambda}^k, \boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)\}$. By taking a subsequence if necessary, without loss of generality, we can suppose that

$$\lim_{k \to \infty} (\boldsymbol{w}^k, \boldsymbol{\lambda}^k, \boldsymbol{\zeta}^k, \boldsymbol{\eta}^k) = (\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\zeta}^*, \boldsymbol{\eta}^*). \tag{28}$$

To show the desired result, it suffices to prove that $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\zeta}^*, \boldsymbol{\eta}^*)$ satisfies (9)–(14). Now, we give the proof by three blocks as follows:

Proof of conditions (9)*,* (10)*,* (13) *and* (14): As for (9) and (10), multiplying (4) and (7) with $(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\zeta}, \boldsymbol{\eta}) = (\boldsymbol{w}^k, \boldsymbol{\lambda}^k, \boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)$ by $\boldsymbol{W}_k^2$ and $\boldsymbol{W}_k$ on the left, respectively, we obtain

$$\begin{aligned} &\boldsymbol{W}_k^2 \nabla f(\boldsymbol{w}^k) + \boldsymbol{W}_k^2 \left(\nabla^2_{\boldsymbol{ww}} G(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k) + \lambda_1^k \nabla^2 \varphi_{\mu_{k-1}}(\boldsymbol{w}^k) \right) \boldsymbol{\zeta}^k = \boldsymbol{W}_k^2 \boldsymbol{\varepsilon}_1^{k-1}, \\ &\boldsymbol{W}_k \nabla_{\boldsymbol{w}} G(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k) + \lambda_1^k \boldsymbol{W}_k \nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k) = \boldsymbol{W}_k \boldsymbol{\varepsilon}_4^{k-1}. \end{aligned}$$

Note that the functions ∇f, $\nabla^2_{\boldsymbol{ww}} G$, and $\nabla_{\boldsymbol{w}} G$ are continuous and let $k \to \infty$ in the above equations. Then, using (21) and (22) in Proposition 6 together with $\mu_{k-1} \to 0$, $(\boldsymbol{\varepsilon}_1^{k-1}, \boldsymbol{\varepsilon}_4^{k-1}) \to (\boldsymbol{0}, \boldsymbol{0})$ as $k \to \infty$, we get (9) and (10), that is to say,

$$\begin{aligned} &\boldsymbol{W}_*^2 \nabla f(\boldsymbol{w}^*) + \boldsymbol{H}(\boldsymbol{w}^*, \boldsymbol{\lambda}^*) \boldsymbol{\zeta}^* = \boldsymbol{0}, \\ &\boldsymbol{W}_* \nabla_{\boldsymbol{w}} G(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*) + p \lambda_1^* |\boldsymbol{w}^*|^p = \boldsymbol{0}. \end{aligned}$$

Conditions (13) and (14) are obtained by driving k to ∞ in (6) and (8) with $(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\zeta}, \boldsymbol{\eta}) = (\boldsymbol{w}^k, \boldsymbol{\lambda}^k, \boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)$.

Proof of condition (12): Choose $i \in I(\boldsymbol{w}^*)$ arbitrarily. Note that the continuity of the functions $\nabla^2_{\boldsymbol{ww}} G$ and ∇f. Then, from (28) and condition (4) with $(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\zeta}) = (\boldsymbol{w}^k, \boldsymbol{\lambda}^k, \boldsymbol{\zeta}^k)$, $\{\lambda_1^k \left(\nabla^2 \varphi_{\mu_{k-1}}(\boldsymbol{w}^k) \right)_{ii} \zeta_i^k\}$ is bounded. On the other hand, recall that $\{(\nabla^2 \varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_{ii}\}$ is unbounded from Proposition 8 and

$\lim_{k\to\infty}\lambda_1^k = \lambda_1^* > 0$ from Assumption A1. Thus, we get $\lim_{k\to\infty}\zeta_i^k = 0$. Since the index i was chosen from $I(\boldsymbol{w}^*)$ arbitrarily, we conclude condition (12).

Proof of condition (11)*:* We begin with proving

$$\lim_{k\to\infty}\sum_{i\in I(\boldsymbol{w}^*)} w_i^k((w_i^k)^2+\mu_{k-1}^2)^{\frac{p}{2}-1}\zeta_i^k = 0. \tag{29}$$

Choose $i \in I(\boldsymbol{w}^*)$ arbitrarily again. Note that by Lemma 7, there exists some $\gamma > 0$ such that

$$\mu_{k-1}^2 \geq \gamma|w_i^k|^{\frac{2}{2-p}} \tag{30}$$

for all k sufficiently large. In what follows, we consider sufficiently large k so that the inequality (30) holds. Then, by $0 < p \leq 1$, we get

$$\frac{\mu_{k-1}^{2-p}}{\gamma^{\frac{2-p}{2}}} \geq |w_i^k|.$$

We then have

$$\begin{aligned}\left|w_i^k((w_i^k)^2+\mu_{k-1}^2)^{\frac{p}{2}-1}\zeta_i^k\right| &\leq \left|w_i^k\mu_{k-1}^{2(\frac{p}{2}-1)}\zeta_i^k\right| \\ &\leq \frac{\mu_{k-1}^{2-p}}{\gamma^{\frac{2-p}{2}}}\mu_{k-1}^{2(\frac{p}{2}-1)}\left|\zeta_i^k\right| \\ &= \gamma^{\frac{p}{2}-1}\left|\zeta_i^k\right|. \end{aligned} \tag{31}$$

Relation (31) and condition (12), which was proved above, imply

$$\lim_{k\to\infty}\left|w_i^k((w_i^k)^2+\mu_{k-1}^2)^{\frac{p}{2}-1}\zeta_i^k\right| = 0$$

and hence summing up this equation over $I(\boldsymbol{w}^*)$ gives the desired expression (29).

Next, by using (28), $\mu_{k-1} \to 0$ $(k \to \infty)$, $w_i^* \neq 0$ $(i \notin I(\boldsymbol{w}^*))$, and $w_i^* = \mathrm{sgn}(w_i^*)|w_i^*|$, we obtain

$$\begin{aligned}&\lim_{k\to\infty}\sum_{i\notin I(\boldsymbol{w}^*)} w_i^k((w_i^k)^2+\mu_{k-1}^2)^{\frac{p}{2}-1}\zeta_i^k \\ &= \sum_{i\notin I(\boldsymbol{w}^*)}\mathrm{sgn}(w_i^*)|w_i^*|^{p-1}\zeta_i^*. \end{aligned} \tag{32}$$

Combining (29) and (32) with (19) yields

$$\begin{aligned}&\lim_{k\to\infty}\nabla\varphi_{\mu_{k-1}}(\boldsymbol{w}^k)^\top\boldsymbol{\zeta}^k \\ &= \lim_{k\to\infty} p\sum_{i=1}^n w_i^k((w_i^k)^2+\mu_{k-1}^2)^{\frac{p}{2}-1}\zeta_i^k \\ &= p\lim_{k\to\infty}\left(\sum_{i\in I(\boldsymbol{w}^*)} w_i^k((w_i^k)^2+\mu_{k-1}^2)^{\frac{p}{2}-1}\zeta_i^k + \sum_{i\notin I(\boldsymbol{w}^*)} w_i^k((w_i^k)^2+\mu_{k-1}^2)^{\frac{p}{2}-1}\zeta_i^k\right) \\ &= p\sum_{i\notin I(\boldsymbol{w}^*)}\mathrm{sgn}(w_i^*)|w_i^*|^{p-1}\zeta_i^*,\end{aligned}$$

which together with driving k to ∞ in condition (5) with $(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\zeta}, \boldsymbol{\eta}) = (\boldsymbol{w}^k, \boldsymbol{\lambda}^k, \boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)$ implies

$$p \sum_{i \notin I(\boldsymbol{w}^*)} \operatorname{sgn}(w_i^*)|w_i^*|^{p-1}\zeta_i^* = \eta_1^*,$$

where we use $\eta_1^* = \lim_{k\to\infty} \eta_1^k$. This is nothing but condition (11). Consequently, the proof of Theorem 5 is complete. ■

Boundedness of the Lagrange-multiplier Sequence

In Assumption **A2**, we suppose that the Lagrange multiplier sequence $\{(\boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)\}$ is bounded. One may ask when this condition holds. In many optimization algorithms, boundedness properties of relevant Lagrange multiplier sequences are shown to be true under suitable constraint qualifications. In fact, we can verify the boundedness of $\{(\boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)\}$ under the presence of linearly constraint-like qualifications as follows:

A4: Let $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*) \in \mathbb{R}^n \times \mathbb{R}^r$ be an arbitrary accumulation point of the sequence $\{(\boldsymbol{w}^k, \boldsymbol{\lambda}^k)\}$. Let

$$I(\boldsymbol{\lambda}^*) := \{i \in \{1, 2, \ldots, r\} \mid \lambda_i^* = 0\}.$$

Then, the linearly independent constraint qualification (LICQ) holds at $(\boldsymbol{w}, \boldsymbol{\lambda}) = (\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ for the constraints $\Phi_i(\boldsymbol{w}, \boldsymbol{\lambda}) := \frac{\partial G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})}{\partial w_i} + p \operatorname{sgn}(w_i)\lambda_1|w_i|^{p-1} = 0 \; (i \notin I(\boldsymbol{w}^*)),\ w_i = 0 \; (i \in I(\boldsymbol{w}^*))$, and $\boldsymbol{\lambda} \geq \mathbf{0}$, that is to say, the gradient vectors for the active constraints

$$\left\{ \{\nabla \Phi_i(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)\}_{i \notin I(\boldsymbol{w}^*)}, \left\{\nabla_{(\boldsymbol{w},\boldsymbol{\lambda})} w_i|_{\boldsymbol{w}=\boldsymbol{w}^*}\right\}_{i\in I(\boldsymbol{w}^*)}, \left\{\nabla_{(\boldsymbol{w},\boldsymbol{\lambda})} \lambda_i|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}^*}\right\}_{i \in I(\boldsymbol{\lambda}^*)} \right\} \subseteq \mathbb{R}^{n+r}$$

are linearly independent.

Proposition 9 *Suppose that Assumptions A1, A3, and A4 hold. Additionally, suppose that the sequence $\{(\boldsymbol{w}^k, \boldsymbol{\lambda}^k)\}$ is bounded. Let $\{(\boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)\} \subseteq \mathbb{R}^n \times \mathbb{R}^r$ be a sequence of the accompanying Lagrange multiplier vectors which satisfy the KKT conditions* (4)–(8). *Then, $\{(\boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)\}$ is bounded.*

Proof We derive contradiction by supposing that $\{(\boldsymbol{\zeta}^k, \boldsymbol{\eta}^k)\}$ is unbounded. For details, see Appendix A.5. ■

5. Numerical Experiments

In this section, we investigate the performance of Algorithm 1 through comparison to other hyperparameter learning methods such as Bayesian optimization (Mockus et al., 1978) and gridsearch. All the experiments are conducted on a personal computer with Intel Core i7-8559U CPU @ 2.70GHz, 16.00 GB memory. Algorithm 1 and the other competitor algorithms are implemented with MATLAB R2020a.

Two kinds of bilevel problems relevant to linear regression with real data are solved. The first problem handles a single hyperparameter related to the ℓ_p-regularizer, while the second one does multiple hyperparameters.

5.1 Linear regression bilevel problem with a single ℓ_p hyperparameter

In this section, we solve the following bilevel problem regarding squared linear regression problem with a single ℓ_p hyperparameter.

$$\begin{array}{ll} \min\limits_{\boldsymbol{w},\boldsymbol{\lambda}} & \|\boldsymbol{A}_{\mathrm{val}}\boldsymbol{w} - \boldsymbol{b}_{\mathrm{val}}\|_2^2 \\ \text{s.t.} & \boldsymbol{w} \in \operatorname*{argmin}\limits_{\hat{\boldsymbol{w}}} \left(\|\boldsymbol{A}_{\mathrm{tr}}\hat{\boldsymbol{w}} - \boldsymbol{b}_{\mathrm{tr}}\|_2^2 + \exp(\lambda_1)\|\hat{\boldsymbol{w}}\|_p^p\right), \end{array} \tag{33}$$

where $p \in (0,1]$ and $(\boldsymbol{A}_{\mathrm{txt}}, \boldsymbol{b}_{\mathrm{txt}}) \in \mathbb{R}^{m_{\mathrm{txt}} \times n} \times \mathbb{R}^{m_{\mathrm{txt}}}$ for $\mathrm{txt} \in \{\mathrm{val}, \mathrm{tr}\}$. Notice that the form of the above problem slightly differs from that of (1) in that λ_i is replaced with $\exp(\lambda_i)$ for each i. This is because positive hyperparameters, in particular λ_1, are actually desirable as outputs. With this manipulation, the nonnegative constraint $\exp(\lambda_i) \geq \mathbf{0}$, which corresponds to $\lambda_i \geq \mathbf{0}$ in (1), is clearly fulfilled and thus removed.

For the sake of examining the accuracy of solutions obtained by solving the above problem, we use $\|\boldsymbol{A}_{\mathrm{te}}\boldsymbol{w} - \boldsymbol{b}_{\mathrm{te}}\|_2^2$ as a test error function, where $\boldsymbol{A}_{\mathrm{te}} \in \mathbb{R}^{m_{\mathrm{te}} \times n}$ and $\boldsymbol{b}_{\mathrm{te}} \in \mathbb{R}^{m_{\mathrm{te}}}$. The data matrices and vectors $\boldsymbol{A}_{\{\mathrm{val,tr,te}\}}, \boldsymbol{b}_{\{\mathrm{val,tr,te}\}}$ are taken from UCI machine learning repository (Lichman et al., 2013): **Facebook** Comment Volume ($\bar{m} = 40949$, $n = 53$), **Insurance** Company Benchmark ($\bar{m} = 9000$, $n = 85$), **Student** Performance for a math exam ($\bar{m} = 395$, $n = 272$)[4], **BodyFat** ($\bar{m} = 336$, $n = 14$), and **CpuSmall** ($\bar{m} = 8192$, $n = 12$) are from UCI machine learning repository Lichman et al. (2013). The $\bar{m}$ samples are divided into 3 groups (training, validation and test samples) with the same sample size $\lceil \bar{m}/3 \rceil$. Hence, $m_{\{\mathrm{val,tr,te}\}} = \lceil \bar{m}/3 \rceil$.

5.1.1 EXPERIMENTAL CONDITIONS

Method for solving the smoothed subproblem (3)**:** Algorithm 1 requires an $\hat{\varepsilon}$-approximate KKT point of (3) in Step 2 at every iteration. To compute such a point, we present an algorithm using implicit functions. Several past works also employed similar approaches based on implicit functions for hyperparameter optimization. For example, see the articles (Maclaurin et al., 2015; Pedregosa, 2016; Franceschi et al., 2018).

We only explain the algorithmic framework of the proposed implicit function based method, leaving the precise description to Algorithm B.1 in Appendix B.1. At every iteration, Algorithm B.1 locally represents problem (3) as problem having the hyperparameter $\boldsymbol{\lambda}$ as variables by means of implicit functions defined over the $\boldsymbol{\lambda}$-space. The implicit function, say $\boldsymbol{w}(\cdot)$, is defined on some open set U and expresses a solution set for the smoothed lower-level problem $\min_{\boldsymbol{w}\in\mathbb{R}^n} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1\varphi_\mu(\boldsymbol{w})$. Namely, we have $\nabla_{\boldsymbol{w}} G(\boldsymbol{w}(\boldsymbol{\lambda}), \bar{\boldsymbol{\lambda}}) + \lambda_1 \nabla\varphi_\mu(\boldsymbol{w}(\boldsymbol{\lambda})) = \mathbf{0}$ for all $\boldsymbol{\lambda} \in U$. We then solve the reformulated problem (3) that is described in terms of $\boldsymbol{\lambda}$ by the quasi-Newton method (Nocedal and Wright, 2006). Note that, though it is difficult in general to know the concrete form of the implicit function $\boldsymbol{w}(\cdot)$, we can compute the gradient $\nabla\boldsymbol{w}$ in virtue of the implicit function theorem, which enables us to perform gradient based methods like the quasi-Newton method for solving problems that are described in terms of $\boldsymbol{w}(\cdot)$.

Actually, the efficiency of this approach relies on how rapidly and accurately $\boldsymbol{w}(\boldsymbol{\lambda})$ is computed for a given $\boldsymbol{\lambda}$. To this end, we employ a certain modified Newton method, which was originally proposed by Lai and Wang (2011). See Appendix B.2 for details.

[4] The original dataset has $n = 32$, but the feature size is increased by adding new features: interaction effects generated by pairwise products among some features for each sample.

Moreover, as a starting point of the algorithm, we use a solution of the smoothed problem (3) at the previous iteration of Algorithm 1, aiming for the so-called hot-start effect.

Other algorithms for comparison: For the sake of comparison, we also implement Bayesian optimization and the gridsearch method. We use `bayesopt` in MATLAB with "MaxObjectiveEvaluations=30" for Bayesian optimization. In gridsearch, we search for the best value of $\|\boldsymbol{A}_{\text{val}}\boldsymbol{w} - \boldsymbol{b}_{\text{val}}\|_2^2$ among 30 grids $\lambda = 10^{-4}, 10^{-4+\frac{8}{29}}, \cdots, 10^{4-\frac{8}{29}}, 10^4$ for problem (33). At each iteration of `bayesopt` and gridsearch, we make use of Matlab built-in solver `fmincon` so as to solve the lower-level problem of (33) with a given $\boldsymbol{\lambda}$.

Parameter setting and termination criteria: The smoothing parameter in Algorithm 1 is initialized as $\mu_0 = 1$ and updated by $\mu_{k+1} = \min(0.9\mu_k, 10\mu_k^{1.3})$. The smoothed subproblem (3) is solved as exactly as possible by fixing $(\hat{\varepsilon}_0, \beta_0)$ to $(10^{-6}, 1)$. As for the termination criteria of Algorithm 1, writing a resulting solution as $\boldsymbol{w}^*$, we stop it if the SB-KKT conditions (9), (10) and (11) are within the error of $\epsilon := 10^{-3}$. We also check whether the other SB-KKT conditions (12)-(14) are satisfied. The default setting of `bayesopt` is employed. Time limits of all the algorithms are set to 600 seconds.

5.1.2 NUMERICAL RESULTS FOR PROBLEM (33) WITH FIXED DATA SIZE

We first show the results of applying the three algorithms to problem (33) with $p = 1, 0.8, 0.5$. The algorithms are run for 5 times from different starting points $(\boldsymbol{\lambda}^0, \boldsymbol{w}^0)$ generated in the manner that $\boldsymbol{\lambda}^0$ is set to $\mathbf{0}$ and $\boldsymbol{w}^0$ is chosen randomly from $[-5, 5]^n$. All the results are summarized in Table 1, where the value Err$_{\text{te}}$ indicates the averaged values of $\|\boldsymbol{A}_{\text{te}}\boldsymbol{w} - \boldsymbol{b}_{\text{te}}\|^2$ over 5 runs. The value Err$_{\text{val}}$ stands for the averaged value of $\|\boldsymbol{A}_{\text{val}}\boldsymbol{w} - \boldsymbol{b}_{\text{val}}\|^2$. The value "sparsity" means the ratio of zero elements in the obtained solution $\boldsymbol{w} \in \mathbb{R}^n$, namely, $\text{sparsity} = |\{i \mid w_i = 0\}| / n$ and hence, the solution with sparsity≈ 1 is very sparse. We denote by time(sec) the spent time from the start to the termination. In the experiments, for each i, we regarded w_i as zero if $|w_i| \leq 10^{-4} \max_{1 \leq i \leq n} |w_i|$. The best (smallest) values of Err$_{\{\text{te,val}\}}$ and time(sec), among the three algorithms are displayed in bold.

From the table, there are significant differences in time(sec) of the three algorithms, while Err$_{\text{te}}$ and Err$_{\text{val}}$ seem comparative. In particular, Algorithm 1 tends to be the fastest. Indeed, it attains the best values in time(sec) for 10 out of 15 problem-instances, seven of which moreover achieve the best values in Err$_{\text{val}}$. For example, for **Facebook** with $p = 1$, it computes a solution with Err$_{\text{val}} = 6.474$ by about 17 seconds, while `bayesopt` and gridsearch do solutions with Err$_{\text{val}} \geq 6.476$ after spending more than 40 seconds. Thus, Algorithm 1 is likely to be the most effective among the three on seeking $(\boldsymbol{w}, \boldsymbol{\lambda})$ with good Err$_{\text{val}}$. As pointed out by a reviewer, `bayesopt` actually found the final solutions or close solutions earlier than the recorded time on the table. Nonetheless, in many instances, Algorithm 1 reached the final solution more quickly than `bayesopt` found such a solution. We refer readers to Table C.1 in Appendix C, which shows the first time of `bayesopt` for finding a solution which attains the final best observed objective value, namely, validation value. Also see Figure C.1 that depicts the time-series of the best observed objective value of `bayesopt` for the problem organized with the data sets of **Student** and **CpuSmall**.

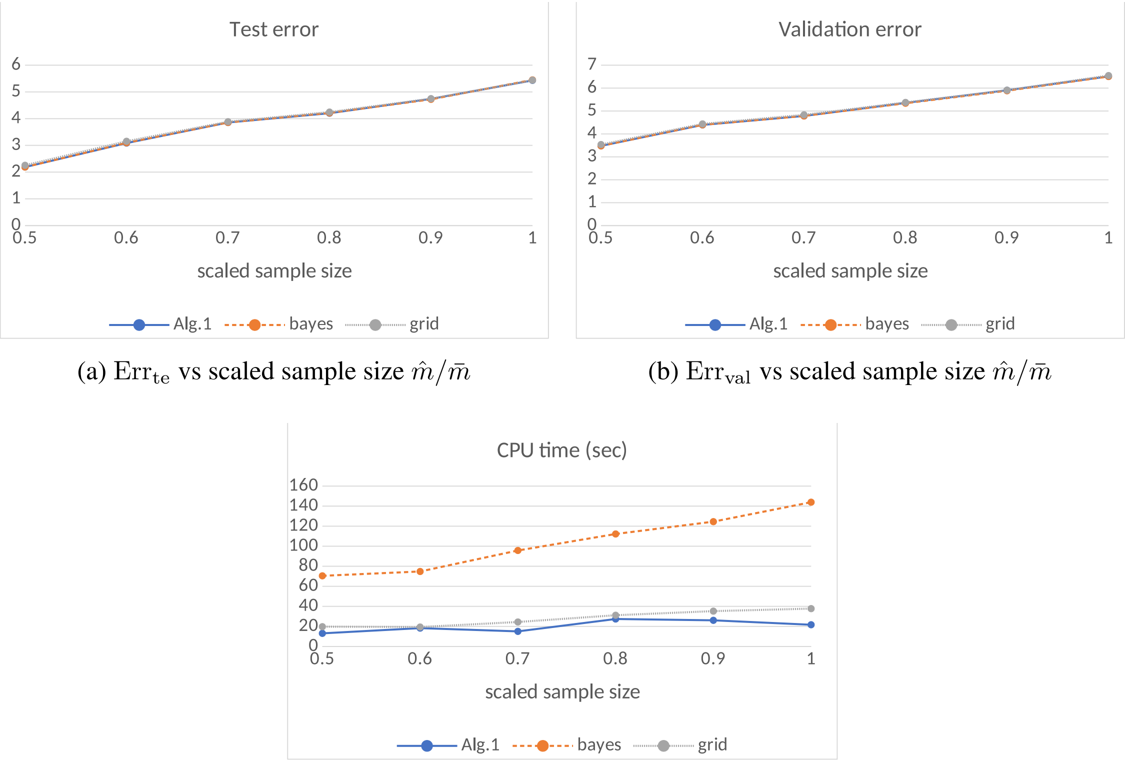
From the values of sparsity, the problems with smaller p tends to output sparser solutions. For example, Algorithm 1 outputs a solution with $\text{sparsity} \geq 0.9$ for **Facebook** with $p = 0.5$, while $\text{sparsity} \leq 0.3$ for $p = 0.8, 1$. Nevertheless, sparsity of Algorithm 1 is 0.00 for **BodyFat** with

$p = 0.8$, while sparsity $= 0.7, 0.07$ for $p = 0.5, 1$, respectively. In this case, Algorithm 1 might fall into a local optimum with small E_{val} at the expense of sparsity.

5.1.3 PERFORMANCE WITH VARIED DATA SIZE

Changing the data sizes of **Student** and **Facebook** datasets, we make comparison of the performances of Algorithm 1, `bayesopt`, and gridsearch.

We first examine how Err_{te}, Err_{val}, and time(sec) of the three algorithms change against the sample size, $\hat{m}$, of **Facebook**. The sample size $\hat{m}$ is increased from $\frac{1}{2}\bar{m}$ to $\bar{m}$ by $\frac{1}{10}\bar{m}$ with $\bar{m} \approx 40000$. We apply the algorithms using $\ell_{0.8}$ regularizer to these problems with a varied sample size. The obtained results are depicted in Figure 1. Figures 1a and 1b indicates that the computed test and validation values Err_{te} and Err_{val} behave analogously as $\hat{m}$ increases. There are no crucial differences among those values. However, from Figure 1c, computation time, time(sec), of `bayesopt` grows more rapidly than the others. This may be because `bayesopt` has to search a wider region as the sample size grows. In contrast, the values of time(sec) of Algorithm 1 and gridsearch grow moderately. In particular, Algorithm 1 is the fastest for most cases.



(a) Err_{te} vs scaled sample size $\hat{m}/\bar{m}$

(b) Err_{val} vs scaled sample size $\hat{m}/\bar{m}$

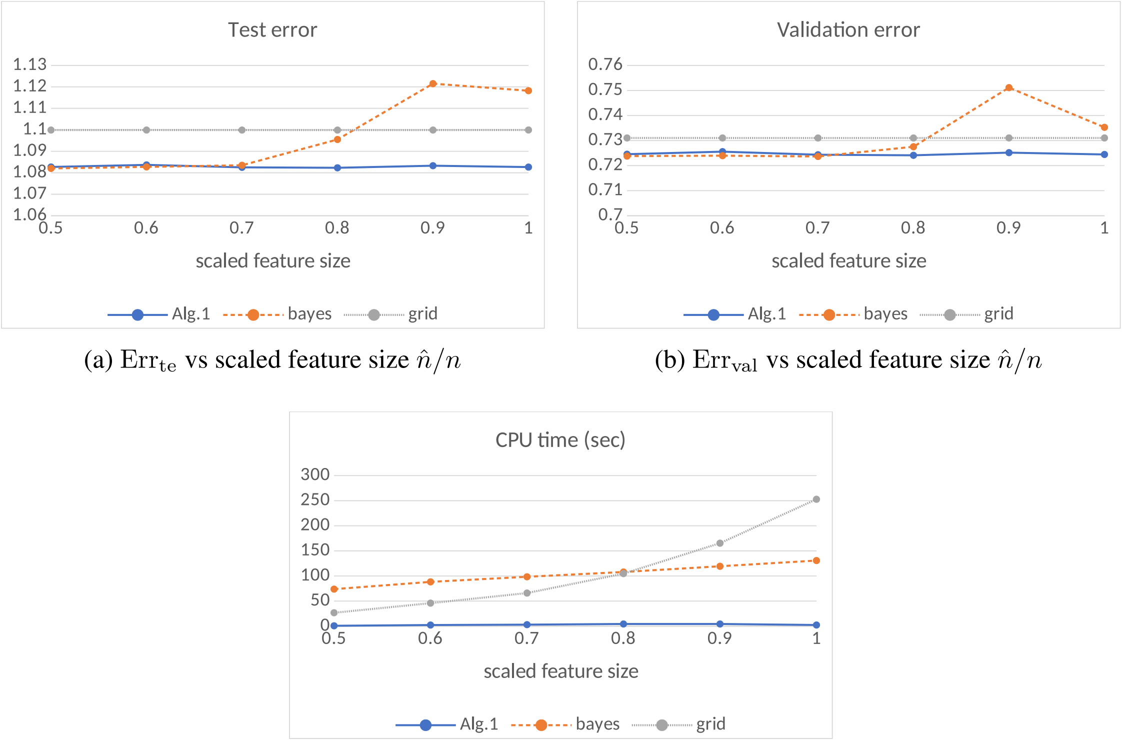
(c) time(sec) vs scaled sample size $\hat{m}/\bar{m}$

Figure 1: Performance of Algorithm 1, `bayesopt`, and gridsearch using $\ell_{0.8}$ regularizer for **Facebook** with fixed feature size $n = 53$ and varying sample size $\hat{m}$ ($\bar{m} = 40949$); Alg.1: Algorithm 1, bayes: `bayesopt`, grid: gridsearch

Next, we investigate the performances of the algorithms by varying the feature size, $\hat{n}$, of **Student**. As in the above experiment, we use $\ell_{0.8}$ regularizer. The feature size $\hat{n}$ is increased from $\frac{1}{2}n$ to

n by $\frac{1}{10}n$ with $n = 272$. The obtained results are shown in Figure 2. Algorithm 1 successfully attains better values in all time(sec), $\mathrm{Err}_{\mathrm{te}}$, and $\mathrm{Err}_{\mathrm{val}}$ than `bayesopt` and gridsearch for $n \geq 0.7\hat{n}$. From Figures 2a and 2b, `bayesopt` seems stuck in a local optimum with larger $\mathrm{Err}_{\mathrm{val}}$ and $\mathrm{Err}_{\mathrm{te}}$ for $n \geq 0.8\hat{n}$. From Figure 2c, time(sec) for `bayesopt` and gridsearch grow more rapidly than ours as $\hat{n}$ increases.

The above two experiments suggest that, compared with gridsearch and Bayesian optimization, Algorithm 1 is unlikely to be affected by growth of the data size.



(a) $\mathrm{Err}_{\mathrm{te}}$ vs scaled feature size $\hat{n}/n$

(b) $\mathrm{Err}_{\mathrm{val}}$ vs scaled feature size $\hat{n}/n$

(c) time(sec) vs scaled feature size $\hat{n}/n$

Figure 2: Performance of Algorithm 1, `bayesopt`, and gridsearch using $\ell_{0.8}$ regularizer for **Student** with varying feature size $\hat{n}$ ($n = 272$) and fixed sample size $\bar{m} = 395$; Alg.1: Algorithm 1, bayes: `bayesopt`, grid: gridsearch

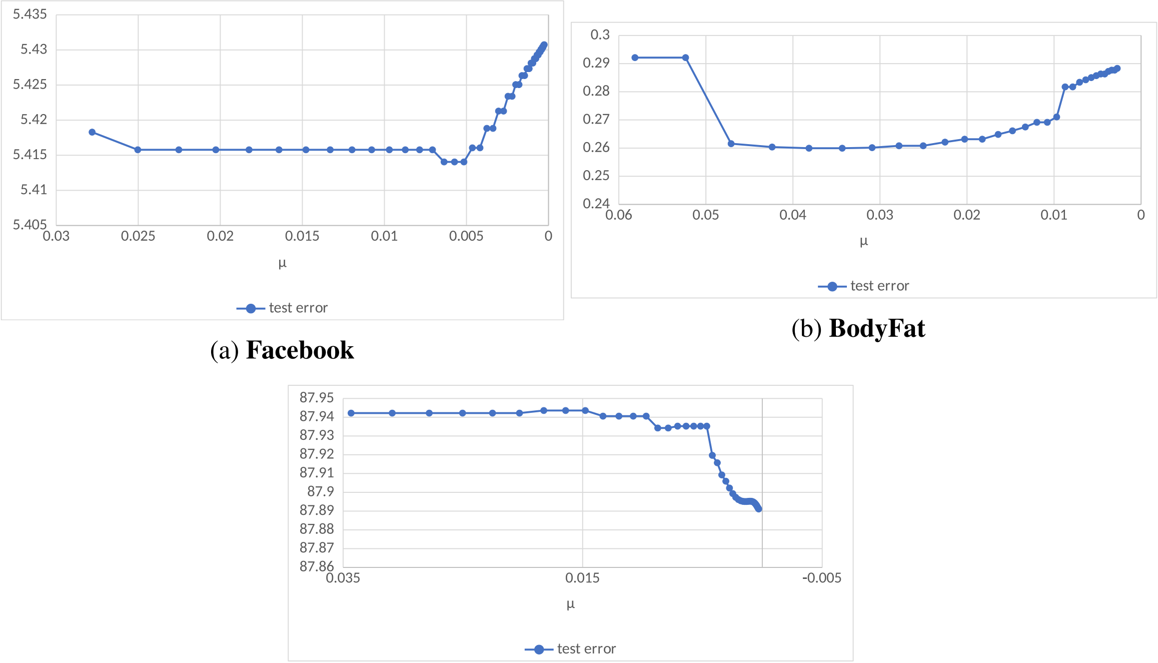
Table 1: Comparison of Algorithm 1, `bayesopt` in MATLAB and gridsearch in terms of squared errors (validation error $\mathrm{Err}_{\mathrm{val}}$ and test error $\mathrm{Err}_{\mathrm{te}}$), CPU times (time(sec)), and sparsities (sparsity). Here, $\mathrm{sparsity} := |\{i \mid w_i = 0\}| / n$ and hence a solution is sparser as "sparsity" is closer to 1. The best values in $\mathrm{Err}_{\mathrm{val}}$, $\mathrm{Err}_{\mathrm{te}}$, and time(sec) are displayed in bold.

Data		Algorithm 1				`bayesopt` in MATLAB				grid search			
name	p	$\mathrm{Err}_{\mathrm{te}}$	$\mathrm{Err}_{\mathrm{val}}$	time (sec)	sparsity	$\mathrm{Err}_{\mathrm{te}}$	$\mathrm{Err}_{\mathrm{val}}$	time (sec)	sparsity	$\mathrm{Err}_{\mathrm{te}}$	$\mathrm{Err}_{\mathrm{val}}$	time (sec)	sparsity
Facebook	1	5.399	**6.474**	**17.399**	0.057	5.401	6.476	146.144	0.249	**5.394**	6.483	43.738	0.264
	0.8	5.431	6.512	**22.242**	0.245	**5.411**	**6.499**	137.764	0.143	5.439	6.545	37.021	0.113
	0.5	5.455	6.550	16.820	0.925	**5.452**	**6.535**	93.967	0.596	5.704	6.747	**16.514**	0.925
Insurance	1	**87.837**	**95.764**	33.077	0.035	87.842	**95.764**	55.304	0.435	87.843	95.765	**19.779**	0.435
	0.8	87.891	95.676	32.465	0.188	87.882	**95.634**	75.384	0.287	**87.872**	95.806	**19.388**	0.329
	0.5	88.625	95.562	44.904	0.859	**88.101**	**95.526**	45.359	0.675	88.211	96.592	**5.453**	0.871
Student	1	**1.127**	**0.778**	**10.586**	0.625	1.147	0.785	217.415	0.032	1.147	0.785	81.766	0.066
	0.8	1.083	**0.724**	**2.348**	0.996	**1.082**	**0.724**	409.382	0.004	1.100	0.731	241.298	0.004
	0.5	**1.082**	**0.724**	**3.618**	0.996	1.125	0.772	152.826	0.829	2.848	1.120	9.520	0.974
BodyFat	1	0.277	**0.209**	**0.068**	0.071	**0.276**	0.210	10.253	0.714	0.279	0.216	0.820	0.714
	0.8	0.288	**0.179**	**0.203**	0.000	**0.280**	0.184	10.567	0.571	0.287	0.182	0.808	0.429
	0.5	0.582	0.267	**0.395**	0.714	0.353	0.229	7.279	0.286	**0.316**	**0.256**	0.931	0.286
CpuSmall	1	132326	**130981**	11.299	0.083	**131834**	131123	21.334	0.917	132261	130983	**1.164**	0.917
	0.8	132339	**130982**	**0.741**	0.250	**131770**	131187	19.909	0.917	132205	130991	1.240	0.750
	0.5	132093	**131058**	**0.672**	0.250	**131754**	131234	17.619	0.917	132127	131059	1.514	0.750

5.1.4 Performance as the smoothing parameter μ decreases

We examine impact of the smoothing parameter μ on test error of solutions of the smoothed subproblems (3). Figures 3a-3c depict the growth behavior of the test error in the final stage of Algorithm 1 for the problems of **Facebook**, **BodyFat**, and **Insurance**.

From the figures, the test errors for the three problems do not vary significantly. Taking into account that the smoothed subproblem (3) is more difficult as μ becomes smaller, it may be good strategy to stop the algorithm earlier than convergence to a SB-KKT point. This is also indicated by the fact that, around $\mu = 0.01$, the algorithm attains solutions with better test errors than the solutions upon termination for **Facebook** and **BodyFat**.



(a) **Facebook**

(b) **BodyFat**

(c) **Insurance**

Figure 3: Change of test error for $\ell_{0.8}$ regularizer as μ decreases; The horizontal axis: a smoothing parameter μ; The vertical axes: test error

5.2 Linear regression problem with multiple hyperparameters

Next, we solve the following problem that possesses multiple hyperparameters:

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{\lambda}} \quad & \|\boldsymbol{A}_{\text{val}}\boldsymbol{w} - \boldsymbol{b}_{\text{val}}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{w} \in \operatorname*{argmin}_{\hat{\boldsymbol{w}}} \left(\|\boldsymbol{A}_{\text{tr}}\hat{\boldsymbol{w}} - \boldsymbol{b}_{\text{tr}}\|_2^2 + \exp(\lambda_1)\|\hat{\boldsymbol{w}}\|_p^p + \hat{\boldsymbol{w}}^\top \boldsymbol{C}(\bar{\boldsymbol{\lambda}})\hat{\boldsymbol{w}} \right), \end{aligned} \tag{34}$$

where $\boldsymbol{C}(\bar{\boldsymbol{\lambda}}) := \mathrm{Diag}(\exp(\lambda_i))_{i=2}^{n+1}$ being positive definite and $\boldsymbol{A}_{\{\text{val,tr,te}\}}$ and $\boldsymbol{b}_{\{\text{val,tr,te}\}}$ are the ones used in the previous experiments in Section 5.1.

5.2.1 EXPERIMENTAL CONDITIONS

We set $p = 0.5$ in problem (34). The experimental conditions are almost the same as those for problem (33). The main differences are as follows: We make comparison of Bayesian Optimization `bayesopt` with "MaxObjectiveEvaluations=300" and Algorithms 1 using the two algorithms for solving subproblems (3): The first one is the implicit function approach as in the previous experiments and the second one is `fmincon`, where we opt for the SQP method and set "MaxIterations= 10^7". Though we also implemented gridsearch seeking a solution over 30^n grids $\boldsymbol{\lambda} \in \{10^{-4}, 10^{-4+\frac{8}{29}}, \cdots, 10^{4-\frac{8}{29}}, 10^4\}^n$, the obtained results were actually quite poor because the number of grids, which was larger than 30^{10}, was too huge to search. We thus omit those results with gridsearch. Finally, according to the observation in the last experiment in Subsection 5.1.4, we terminate Algorithm 1 when $\mu_k \leq 0.01$.

5.2.2 NUMERICAL RESULTS FOR PROBLEM (34)

Table 2 summarizes the obtained results of applying the two types of Algorithms 1 and `bayesopt` to problem (34). In the table, we denote by $\sharp\boldsymbol{\lambda}$ the number of hyperparameters $\boldsymbol{\lambda}$ in each problem. Moreover, Alg.1-A stands for Algorithm 1 using the implicit function approach and Alg.1-B does the one using `fmincon`. The hyphens "–" in the line of **Facebook** for Alg.1-B indicate that `fmincon`, which is used in Alg.1-B, terminates with an infeasible solution of the smoothed problem (3).

From the table, compared with the results for the single-hyperparameter bilevel problem (33), `bayesopt` does not work well. For four out of the five problems, it cannot terminate within the time-limit 600 seconds. The qualities of the output solutions of `bayesopt` upon termination are also not good in values of $\mathrm{Err}_{\mathrm{val}}$ and $\mathrm{Err}_{\mathrm{te}}$. For the sake of completeness, as well as the previous experiment, we examined the first time when the best observed objective values, that is, validation values of `bayesopt` were found. Refer to Table C.2 in Appendix C. Also see Figure C.2 for graphs depicting the time-series of the best observed objective values concerning the data sets of **Student** and **CpuSmall**.

In contrast to `bayesopt`, the two Algorithms 1 with different subroutines show better performance. There are notable differences between performances of Alg.1-A and Alg.1-B. While Alg.1-A seems to fall into non-sparse solutions, but with good $\mathrm{Err}_{\{\mathrm{te},\mathrm{val}\}}$ for **Insurance**, **BodyFat**, and **CpuSmall**, Alg.1-B finds good solutions balancing in sparsity, $\mathrm{Err}_{\mathrm{te}}$, and $\mathrm{Err}_{\mathrm{val}}$ for all the same data sets. This may be because Alg.1-A computes a solution of problem (3) by remaining in the feasible set, namely, the solution set of the smoothed lower level problem. This behavior may cause Alg.1-A to miss a chance of broadly seeking sparse solutions. Meanwhile, Alg.1-B using the SQP method in `fmincon` tends to approach a solution of (3) from the outside of the feasible set, which often leads to a good solution even in sparsity.

Table 2: Comparison of `bayesopt` and Algorithm s 1 using the implicit function approach and `fmincon` as subroutines for solving subproblem (3) in Step 2 in terms of squared errors (validation error $\mathrm{Err}_{\mathrm{val}}$ and test error $\mathrm{Err}_{\mathrm{te}}$), CPU times (time(sec)), and sparsities (sparsity). Here, $\mathrm{sparsity} := |\{i \mid w_i = 0\}| / n$ and hence a solution is sparser as "sparsity" is closer to 1. Alg.1-A stands for Algorithm 1 using the implicit function approach and Alg.1-B does the one using `fmincon`. The notation $\sharp\boldsymbol{\lambda}$ stands for the number of hyperparameters in each problem. Moreover, the best values in $\mathrm{Err}_{\mathrm{val}}$, $\mathrm{Err}_{\mathrm{te}}$, and time(sec) are displayed in bold. The hyphens "–" for **Facebook** indicate that `fmincon`, which is used in Alg.1-B, terminates with an infeasible solution of the smoothed problem (3). The algorithms with "600" seconds stopped at the time-limit.

Data		Alg.1-A				Alg.1-B				`bayesopt` in MATLAB			
name	$\sharp\boldsymbol{\lambda}$	$\mathrm{Err}_{\mathrm{te}}$	$\mathrm{Err}_{\mathrm{val}}$	time (sec)	sparsity	$\mathrm{Err}_{\mathrm{te}}$	$\mathrm{Err}_{\mathrm{val}}$	time (sec)	sparsity	$\mathrm{Err}_{\mathrm{te}}$	$\mathrm{Err}_{\mathrm{val}}$	time (sec)	sparsity
Facebook	54	**5.404**	6.478	20.247	0.038	–	–	–	–	7.629	8.780	600.000	0.000
Insurance	86	**87.920**	95.694	50.714	0.000	88.340	**94.604**	**4.473**	0.988	98.000	107.000	600.000	0.000
Student	273	**1.132**	**0.771**	**1.451**	0.368	1.142	0.786	71.775	0.670	21.002	18.324	600.000	0.000
BodyFat	15	0.531	0.243	**0.072**	0.000	**0.286**	**0.130**	0.695	0.933	46.675	48.815	455.230	0.000
CpuSmall	13	132780	131130	6.222	0.000	**131940**	**128540**	**0.658**	0.692	156420	151670	600.000	1.000

6. Discussion on extension to other nonsmooth regularizers

In this section, we discuss the extension of the SB-KKT conditions from hyperparameter optimization of ℓ_p-regularizers to those of other nonsmooth and nonconvex regularizers. Let $\Theta : [0, \infty) \to [0, \infty)$ be a function such that it is concave and continuously differentiable on $(0, \infty)$ and $\Theta(0) = 0$. Many sparse regularizers are representable in terms of Θ. Indeed, $\sum_{i=1}^{n} \Theta(|w_i|)$ reduces to the ℓ_p- and log-regularizers, SCAD, and MCP by selecting Θ appropriately as follows. Here, $a > 1$, $b > 0$, and $\gamma > 0$ are prefixed parameters and $x \geq 0$. The continuous differentiability of Θ for SCAD and MCP can be confirmed in view of the formula of Θ':

- ℓ_p-regularizer $(p \leq 1)$ if $\Theta(x) := x^p$;
- log-regularizer (Candes et al., 2008) if $\Theta(x) := \dfrac{1}{\log(1+\gamma)} \log(1 + \gamma x)$;
- SCAD (Fan and Li, 2001) if

$$\Theta(x) := \begin{cases} bx & \text{if } x \leq b \\ -\dfrac{x^2 - 2abx + b^2}{2(a-1)} & \text{if } b \leq x \leq ab \\ \dfrac{(a+1)b^2}{2} & \text{otherwise.} \end{cases}$$

  The first-order derivative is

$$\Theta'(x) = \begin{cases} b & \text{if } x \leq b \\ -\dfrac{x - ab}{a-1} & \text{if } b \leq x \leq ab \\ 0 & \text{otherwise;} \end{cases}$$

- MCP (Zhang et al., 2010) if

$$\Theta(x) := \begin{cases} bx - \dfrac{x^2}{2a} & \text{if } x \leq ab \\ \dfrac{ab^2}{2} & \text{otherwise.} \end{cases}$$

  The first-order derivative is

$$\Theta'(x) = \begin{cases} b - \dfrac{x}{a} & \text{if } x \leq ab \\ 0 & \text{otherwise.} \end{cases}$$

Consider the following extended formulation from problem (1):

$$\min_{\boldsymbol{w}^*_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}} f(\boldsymbol{w}^*_{\boldsymbol{\lambda}}) \ \text{ s.t. } \ \boldsymbol{w}^*_{\boldsymbol{\lambda}} \in \operatorname*{argmin}_{\boldsymbol{w} \in \mathbb{R}^n} \left(G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 \sum_{i=1}^{n} \Theta(|w_i|) \right), \ \boldsymbol{\lambda} \geq \boldsymbol{0}.$$

Any local optimum $\boldsymbol{w}$ of the lower-level problem in the above satisfies

$$\tfrac{\partial G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})}{\partial w_i} + \operatorname{sgn}(w_i) \lambda_1 \Theta'(|w_i|) = 0 \ \text{ for } i \in \{1, 2, \ldots, n\} \setminus I(\boldsymbol{w}), \tag{35}$$

$$w_i = 0 \ \text{ for } i \in I(\boldsymbol{w}), \tag{36}$$

which actually correspond with the scaled first order condition (16) after transformations when Θ is chosen to be the ℓ_p-regularizer. We then obtain the following one-level problem that corresponds to (17)

$$\min_{\boldsymbol{w},\boldsymbol{\lambda}} \; f(\boldsymbol{w}) \;\; \text{s.t.} \;\; \boldsymbol{\lambda} \geq \mathbf{0}, \; \boldsymbol{w} \text{ satisfies (35) and (36).} \tag{37}$$

The following theorem states the SB-KKT conditions for (37), which are the extended version of Theorem 2. Since its proof can be obtained via straightforward extension of that of Theorem 2 by noting the relation $(\Theta(|x|), \Theta'(|x|), \Theta''(|x|)) = \left(|x|^p, \mathrm{sgn}(x)p|x|^{p-1}, p(p-1)|x|^{p-2}\right)$ for $x \neq 0$ when $\Theta(x) = x^p$ for $x \geq 0$, we omit it.

Theorem 10 *Let $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*) \in \mathbb{R}^n \times \mathbb{R}^r$ be a local optimum of (37). Suppose that Θ is twice continuously differentiable at $|w_i^*|$ for $i \notin I(\boldsymbol{w}^*)$. Then, $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ together with some vectors $\boldsymbol{\zeta}^* \in \mathbb{R}^n$ and $\boldsymbol{\eta}^* \in \mathbb{R}^r$ satisfies the following conditions under an appropriate constraint qualification concerning the constraints $\frac{\partial G(\boldsymbol{w},\bar{\boldsymbol{\lambda}})}{\partial w_i} + \lambda_1 \mathrm{sgn}(w_i)\Theta'(|w_i|) = 0$ $(i \notin I(\boldsymbol{w}^*))$, $w_i = 0$ $(i \in I(\boldsymbol{w}^*))$, and $\boldsymbol{\lambda} \geq \mathbf{0}$:*

$$\nabla f(\boldsymbol{w}^*) + \textstyle\sum_{i \notin I(\boldsymbol{w}^*)} \boldsymbol{H}_i(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)\zeta_i^* = \mathbf{0}, \tag{38}$$

$$\frac{\partial G(\boldsymbol{w}^*,\bar{\boldsymbol{\lambda}}^*)}{\partial w_i} + \lambda_1^* \mathrm{sgn}(w_i^*)\Theta'(|w_i^*|) = 0 \; (i \notin I(\boldsymbol{w}^*)), \tag{39}$$

$$\begin{gathered} \textstyle\sum_{i \notin I(\boldsymbol{w}^*)} \mathrm{sgn}(w_i^*)\Theta'(|w_i^*|)\zeta_i^* = \eta_1^*, \\ \zeta_i^* = 0 \; (i \in I(\boldsymbol{w}^*)), \\ \nabla R_i(\boldsymbol{w}^*)^\top \boldsymbol{\zeta}^* = \eta_i^* \;\; (i = 2, 3, \ldots, r), \\ \mathbf{0} \leq \boldsymbol{\lambda}^*, \; \mathbf{0} \leq \boldsymbol{\eta}^*, \; (\boldsymbol{\lambda}^*)^\top \boldsymbol{\eta}^* = 0, \end{gathered} \tag{40}$$

where

$$\boldsymbol{H}_i(\boldsymbol{w}, \boldsymbol{\lambda}) := \nabla_{\boldsymbol{w}} \left(\frac{\partial G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})}{\partial w_i} \right) + \lambda_1 \mathrm{sgn}(w_i)\Theta''(|w_i|)\boldsymbol{e}^i \in \mathbb{R}^n \;\; (i \notin I(\boldsymbol{w}^*)).$$

When $\Theta(x) = x^p$ with $x \geq 0$ and $0 < p \leq 1$, conditions (38) and (39) premultiplied by $\mathrm{diag}(\boldsymbol{w}^*)^2$ and $\mathrm{diag}(\boldsymbol{w}^*)$, respectively, are equivalent to (9) and (10) under the presence of (40) and $w_i^* = 0$ $(i \in I(\boldsymbol{w}^*))$. The above theorem is different from Theorem 2 in that Θ is additionally assumed to be C^2 at $|w_i^*|$ for $i \notin I(\boldsymbol{w}^*)$. This is due to the existence of the term Θ'' in $\boldsymbol{H}_i$. If Θ is chosen to correspond to ℓ_p or log-regularizer, this assumption always holds. In contrast, if Θ is selected to correspond to SCAD (resp., MCP), it is equivalent to $w_i^* \neq b, ab$ (resp., $w_i^* \neq ab$) for $i \notin I(\boldsymbol{w}^*)$ and thus may fail to hold in general. Though it is expected to hold in many instances, we need to do a further research so as to remove or weaken it.

It is easy to tailor the proposed smoothing method to problems having other regularizers such as SCAD and MCP. Convergence properties similar to the case of using the ℓ_p-regularizer hold expectedly. However, proofs for the global convergence to an SB-KKT point in the sense of Theorem 10 may differ significantly from that in Section 4, because our analysis for the ℓ_p-regularizer actually relies on the specific forms of the smoothing function $\varphi_\mu(\boldsymbol{w}) = \sum_{i=1}^n (w_i^2 + \mu^2)^{\frac{p}{2}}$ and its first- and second-order derivatives.

Further extension: Besides the above, there are other directions for extending our results. One direction is extension to structured sparse regularizers like the group Lasso model (Yuan and Lin, 2006). Such a model often contains regularizers of the composite form $\sum_{i=1}^l \Theta(\theta_i(\boldsymbol{w}))$ with

$\theta_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, 2, \ldots, l)$. For instances of θ_i, we can set $\theta_i(\boldsymbol{w}) := \boldsymbol{w}^\top \boldsymbol{a}^i$ with $\boldsymbol{a}^i \in \mathbb{R}^n$ or $\theta_i(\boldsymbol{w}) := \boldsymbol{w}^\top \boldsymbol{K}_i \boldsymbol{w}$ with $\boldsymbol{K}_i \in \mathbb{R}^{n\times n}$ being a symmetric positive definite matrix.

Another interesting direction is extension to problems with matrix variables. Marjanovic and Solo (2012) considered regularized least square optimization for matrix completion. For $\boldsymbol{X} \in \mathbb{R}^{r_1 \times r_2}$, the regularization term that appears there takes the form of $\|\boldsymbol{X}\|_p^p := \sum_{i=1}^{\min(r_1,r_2)} \sigma_i(\boldsymbol{X})^p$ with $0 < p \le 1$, where $\sigma_i(\boldsymbol{X})$ $(i = 1, 2, \ldots, \min(r_1, r_2))$ are the singular values of X. If $r_1 = r_2$ and X is a diagonal matrix, $\|\boldsymbol{X}\|_p^p$ reduces to the ℓ_p-regularizer we have considered. In (Marjanovic and Solo, 2012), in order to find the best-qualified recovered matrix model, the authors iteratively solved problems involving $\|\boldsymbol{X}\|_p^p$ as a regularizer while varying hyperparameters. A bilevel approach may help to recover a matrix with higher quality faster.

7. Conclusions

We have proposed a bilevel optimization approach for selecting the best hyperparameter (regularization parameter) of the ℓ_p-regularizer. The bilevel optimization problem that appears in our approach has a nonsmooth and possibly nonconvex ℓ_p-regularized problem as the lower-level problem. For this problem, we have developed the scaled bilevel KKT (SB-KKT) conditions and proposed a smoothing-type method. Furthermore, we have made analysis on convergence of the proposed algorithm to an SB-KKT point. Numerical experiments imply that it exhibited performance superior to Bayesian optimization and grid search especially in computational time.

The method/theoretical guarantee can be applicable to hyperparameter learning for classification. As a future work, we would like to make the algorithm more practical. For this purpose, we may need to integrate some stochastic technique into the proposed algorithm. For example, approximate KKT points computed by approximate gradient and Hessians can be used. In the stochastic setting, we expect that the SB-KKT conditions will play a significant role in convergence analysis.

Acknowledgments

We thank three anonymous reviewers and the action editor Tomioka for their valuable comments and suggestions. This research was supported by JSPS KAKENHI Grant Numbers 20K19748 and 19H04069 and by JST ERATO Grant Number JPMJER1903.

Appendix A. Omitted Proofs

In this section, we provide proofs of some lemmas and propositions.

A.1 Proof of Theorem 2

Firstly, notice that $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ is also a local optimum of the following problem:

$$\begin{array}{cc} \displaystyle\min_{\boldsymbol{w},\boldsymbol{\lambda}} & f(\boldsymbol{w}) \\ \text{s.t.} & \dfrac{\partial G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})}{\partial w_i} + p\,\mathrm{sgn}(w_i)\lambda_1 |w_i|^{p-1} = 0 \ (i \notin I(\boldsymbol{w}^*)) \\ & w_i = 0 \ (i \in I(\boldsymbol{w}^*)) \\ & \boldsymbol{\lambda} \geq \mathbf{0}. \end{array} \tag{A.1}$$

Actually, this fact is easily confirmed by noting that $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ is also feasible to (A.1) and the feasible region of (17) is larger than that of (A.1). Hence, under an appropriate constraint qualification such as the linearly independent constraint qualification associated to (A.1), the KKT conditions for (A.1) hold at $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$, namely, there exist some vectors $\hat{\boldsymbol{\zeta}}^* := (\hat{\zeta}_1^*, \hat{\zeta}_2^*, \dots, \hat{\zeta}_n^*)^\top \in \mathbb{R}^n$ and $\boldsymbol{\eta}^* \in \mathbb{R}^r$ such that

$$\frac{\partial f(\boldsymbol{w}^*)}{\partial w_i} + \sum_{j \notin I(\boldsymbol{w}^*)} \left(\frac{\partial^2 G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})}{\partial w_i \partial w_j} + p(p-1)\lambda_1 |w_i|^{p-2} \right) \hat{\zeta}_j^* = 0 \ \ (i \notin I(\boldsymbol{w}^*)), \tag{A.2}$$

$$\frac{\partial f(\boldsymbol{w}^*)}{\partial w_i} + \sum_{j \notin I(\boldsymbol{w}^*)} \frac{\partial^2 G(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*)}{\partial w_i \partial w_j} \hat{\zeta}_j^* + \hat{\zeta}_i^* = 0 \ \ (i \in I(\boldsymbol{w}^*)),$$

$$\nabla_{\boldsymbol{\lambda}} f(\boldsymbol{w}^*) - \boldsymbol{\eta}^* + \sum_{i \notin I(\boldsymbol{w}^*)} \frac{\partial}{\partial \boldsymbol{\lambda}} \left(\frac{\partial G(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*)}{\partial w_i} + p \operatorname{sgn}(w_i) \lambda_1 |w_i|^{p-1} \right) \hat{\boldsymbol{\zeta}}_i^* = \boldsymbol{0}, \tag{A.3}$$

$$\frac{\partial G(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*)}{\partial w_i} + p \operatorname{sgn}(w_i^*) \lambda_1^* |w_i^*|^{p-1} = 0 \ \ (i \notin I(\boldsymbol{w}^*)), \tag{A.4}$$

$$w_i^* = 0 \quad (i \in I(\boldsymbol{w}^*)), \tag{A.5}$$

$$\boldsymbol{0} \leq \boldsymbol{\lambda}^*, \ \boldsymbol{0} \leq \boldsymbol{\eta}^*, \ (\boldsymbol{\lambda}^*)^\top \boldsymbol{\eta}^* = 0, \tag{A.6}$$

where $\hat{\zeta}_i^*$ $(i \in I(\boldsymbol{w}^*))$, $\hat{\zeta}_i^*$ $(i \notin I(\boldsymbol{w}^*))$, and $\boldsymbol{\eta}^*$ are Lagrange multipliers corresponding to the constraints $w_i = 0$ $(i \in I(\boldsymbol{w}^*)$ and $\frac{\partial G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})}{\partial w_i} + p \operatorname{sgn}(w_i) \lambda_1 |w_i|^{p-1} = 0$ $(i \notin I(\boldsymbol{w}^*))$, and $\boldsymbol{\lambda} \geq \boldsymbol{0}$, respectively. To derive the first equality above, we made use of the fact

$$\frac{\partial |w_i|^{p-1}}{\partial w_i} = (p-1)\operatorname{sgn}(w_i)|w_i|^{p-2}$$

at $w_i \neq 0$. Noting the relations $\nabla_{\boldsymbol{\lambda}} f(\boldsymbol{w}) = \boldsymbol{0}$, $\partial G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})/\partial \lambda_1 = 0$, $\partial^2 G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})/\partial \lambda_i \partial w_i = \partial R_i(\boldsymbol{w}, \boldsymbol{\lambda})/\partial w_i$ $(i = 2, 3, \dots, r)$, we can rewrite condition (A.3) as

$$\sum_{i \notin I(\boldsymbol{w}^*)} p \operatorname{sgn}(w_i^*) |w_i^*|^{p-1} \hat{\zeta}_i^* = \eta_1^*, \tag{A.7}$$

$$\sum_{j \notin I(\boldsymbol{w}^*)} \frac{\partial R_i(\boldsymbol{w})}{\partial w_j} \hat{\zeta}_j^* = \eta_i^* \ \ (i = 2, \dots, r). \tag{A.8}$$

Next, define $\boldsymbol{\zeta}^* \in \mathbb{R}^n$ as the vector with $\zeta_i^* = 0$ $(i \in I(\boldsymbol{w}^*))$ and $\zeta_i^* = \hat{\zeta}_i^*$ $(i \notin I(\boldsymbol{w}^*))$. Let us show that $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\zeta}^*, \boldsymbol{\eta}^*)$ satisfies the targeted conditions (9)–(14). For each $i \in \{1, 2, \dots, n\}$, we have

$$\begin{aligned}
&(w_i^*)^2 \frac{\partial f(\boldsymbol{w}^*)}{\partial w_i} + (w_i^*)^2 \sum_{j=1}^{n} \frac{\partial^2 G(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*)}{\partial w_i \partial w_j} \zeta_j^* + \lambda_1^* p(p-1)|w_i^*|^p \zeta_i^* \\
&= (w_i^*)^2 \frac{\partial f(\boldsymbol{w}^*)}{\partial w_i} + (w_i^*)^2 \sum_{j \notin I(\boldsymbol{w}^*)} \frac{\partial^2 G(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*)}{\partial w_i \partial w_j} \zeta_j^* + \lambda_1^* p(p-1)|w_i^*|^p \zeta_i^* \\
&= 0,
\end{aligned}$$

where the first equality follows from $\zeta_j^* = 0$ $(j \in I(\boldsymbol{w}^*))$ and the second one can be proved by cases; when $i \in I(\boldsymbol{w}^*)$, the desired equality is obviously true because of (A.5); when $i \notin I(\boldsymbol{w}^*)$, it is obtained from multiplying (A.2) by $(w_i^*)^2$ and using $\zeta_i^* = \hat{\zeta}_i^*$. Therefore, we confirm (9). Similarly, we can deduce (10) and (13) from (A.4) and (A.8) along with the definition of $\boldsymbol{\zeta}^*$, respectively. The remaining conditions (11), (12), and (14) are derived from (A.7), $\zeta_i^* = 0$ $(i \in I(\boldsymbol{w}^*))$, and (A.6), respectively. Putting all the above results together, we confirm that $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\zeta}^*, \boldsymbol{\eta}^*)$ satisfies (9)–(14). Consequently, we have the desired result. ■

A.2 Proof of Lemma 3

We will give a proof of Lemma 3. Firstly, we review the definition and several properties for a subgradient of a given function from Rockafellar and Wets (2009). We finally give a proof of Lemma 3.

Let us define regular and general subgradients for a given function according to Definition 8.3(a),(b) of Rockafellar and Wets (2009). For simplicity, we confine ourselves to a continuous function $f : \mathbb{R}^n \to \mathbb{R}$.

Definition A.1 *For vectors $\boldsymbol{v} \in \mathbb{R}^n$ and $\bar{\boldsymbol{x}} \in \mathbb{R}^n$,*

1. *we say that $\boldsymbol{v}$ is a regular subgradient of f at $\bar{\boldsymbol{x}}$, written $\boldsymbol{v} \in \hat{\partial}_{\boldsymbol{x}} f(\bar{\boldsymbol{x}})$, if $f(\boldsymbol{x}) \geq f(\bar{\boldsymbol{x}}) + \boldsymbol{v}^\top(\boldsymbol{x} - \bar{\boldsymbol{x}}) + o(\|\boldsymbol{x} - \bar{\boldsymbol{x}}\|)$.*
2. *We say that $\boldsymbol{v}$ is a (general) subgradient of f at $\bar{\boldsymbol{x}}$, written $\boldsymbol{v} \in \partial_{\boldsymbol{x}} f(\bar{\boldsymbol{x}})$, if there are sequences $\{\boldsymbol{x}^\nu\} \subseteq \mathbb{R}^n$ converging to $\bar{\boldsymbol{x}}$ and $\{\boldsymbol{v}^\nu\} \subseteq \mathbb{R}^n$ converging to $\boldsymbol{v}$ such that $\boldsymbol{v}^\nu \in \hat{\partial} f(\boldsymbol{x}^\nu)$ for each ν.*

We often simply write $\hat{\partial}_{\boldsymbol{x}}$ and $\partial_{\boldsymbol{x}}$ as $\hat{\partial}$ and ∂, respectively.

Obviously, it holds that $\hat{\partial} f(\boldsymbol{x}) \subseteq \partial f(\boldsymbol{x})$.

The following propositions are useful:

Proposition A.2 *(Rockafellar and Wets, 2009, 8.8(c) Exercise) Let $f_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 0, 1)$ be continuous. Let $f := f_0 + f_1$. If f_0 is continuously differentiable around $\bar{\boldsymbol{x}}$, then $\hat{\partial} f(\bar{\boldsymbol{x}}) = \nabla f_0(\bar{\boldsymbol{x}}) + \hat{\partial} f_1(\bar{\boldsymbol{x}})$ and $\partial f(\bar{\boldsymbol{x}}) = \nabla f_0(\bar{\boldsymbol{x}}) + \partial f_1(\bar{\boldsymbol{x}})$.*

Proposition A.3 *(Rockafellar and Wets, 2009, 8.5 Proposition) Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuous. Then, $\boldsymbol{v} \in \hat{\partial} f(\boldsymbol{x})$ if and only if, on some neighborhood of $\bar{\boldsymbol{x}}$, there exists a differentiable function $g : \mathbb{R}^n \to \mathbb{R}$ such that $\nabla g(\bar{\boldsymbol{x}}) = \boldsymbol{v}$, $g(\boldsymbol{x}) \leq f(\boldsymbol{x})$, and $g(\bar{\boldsymbol{x}}) = f(\bar{\boldsymbol{x}})$. Moreover, g can be taken to be continuously differentiable with $g(\boldsymbol{x}) < f(\boldsymbol{x})$ for all $\boldsymbol{x} \neq \bar{\boldsymbol{x}}$ near $\bar{\boldsymbol{x}}$.*

We next prove the following proposition associated with $\|\boldsymbol{x}\|_p^p$ $(0 < p \leq 1)$.

Proposition A.4 *For $\boldsymbol{x} \in \mathbb{R}^n$, let $I(\boldsymbol{x}) := \{i \mid x_i = 0\}$ and $g(\boldsymbol{x}) := \lambda \|\boldsymbol{x}\|_p^p$ with $0 < p \leq 1$ and $\lambda \geq 0$. Then, for $0 < p < 1$ and $\bar{\boldsymbol{x}} \in \mathbb{R}^n$, we have*

$$\partial g(\bar{\boldsymbol{x}}) = \left\{ \boldsymbol{v} \mid v_i = \lambda p \operatorname{sgn}(\bar{x}_i) |\bar{x}_i|^{p-1} \ (i \notin I(\bar{\boldsymbol{x}})),\ v_i \in \mathbb{R} \ (i \in I(\bar{\boldsymbol{x}})) \right\}. \tag{A.9}$$

On the other hand, for $p = 1$, we have

$$\partial g(\bar{\boldsymbol{x}}) = \left\{ \boldsymbol{v} \mid v_i = \lambda \operatorname{sgn}(\bar{x}_i) \ (i \notin I(\bar{\boldsymbol{x}})), v_i \in [-\lambda, \lambda] \ (i \in I(\bar{\boldsymbol{x}})) \right\}. \tag{A.10}$$

Proof For convenience of expression, let $\hat{g}(\boldsymbol{x}) := \lambda \sum_{i \in I(\bar{\boldsymbol{x}})} |x_i|^p$. Note that $g(\boldsymbol{x}) = \hat{g}(\boldsymbol{x}) + \lambda \sum_{i \notin I(\bar{\boldsymbol{x}})} |x_i|^p$ and $\lambda \sum_{i \notin I(\bar{\boldsymbol{x}})} |x_i|^p$ is continuously differentiable around $\bar{\boldsymbol{x}}$. Then, by Proposition A.2, we have

$$\partial g(\bar{\boldsymbol{x}}) = \lambda \sum_{i \notin I(\bar{\boldsymbol{x}})} p \operatorname{sgn}(\bar{x}_i)|\bar{x}_i|^{p-1}\boldsymbol{e}^i + \partial \hat{g}(\bar{\boldsymbol{x}}), \tag{A.11}$$

where $\boldsymbol{e}^i \in \mathbb{R}^n$ is the vector such that the i-th element is one and the others are zeros. Supposing $I(\bar{\boldsymbol{x}}) \neq \emptyset$, we next describe $\partial_{\boldsymbol{x}} \hat{g}(\bar{\boldsymbol{x}})$ precisely. First, consider the case of $0 < p < 1$. For any $\boldsymbol{v} \in \mathbb{R}^n$ with $v_i = 0$ $(i \notin I(\bar{\boldsymbol{x}}))$, we can show that $\lambda \sum_{i \in I(\bar{\boldsymbol{x}})} |x_i|^p \geq \lambda \sum_{i \in I(\bar{\boldsymbol{x}})} v_i x_i$ holds on a sufficiently small neighborhood of $\bar{\boldsymbol{x}}$ since $\lambda \geq 0$. Then, Proposition A.3 implies

$$\hat{\partial} \hat{g}(\bar{\boldsymbol{x}}) \supseteq \{\boldsymbol{v} \mid v_i = 0 \ (i \notin I(\bar{\boldsymbol{x}}))\}. \tag{A.12}$$

We next show the converse implication for the above. To this end, choose a regular subgradient $\boldsymbol{v} \in \hat{\partial} \hat{g}(\bar{\boldsymbol{x}}) = \hat{\partial} \left(\lambda \sum_{i \in I(\bar{\boldsymbol{x}})} |x_i|^p\right)\Big|_{\boldsymbol{x}=\bar{\boldsymbol{x}}}$ arbitrarily. Then, according to Proposition A.3, there exists some differentiable function h such that $h(\boldsymbol{x}) \leq \lambda \sum_{i \in I(\bar{\boldsymbol{x}})} |x_i|^p$ near $\bar{\boldsymbol{x}}$, $h(\bar{\boldsymbol{x}}) = \lambda \sum_{i \in I(\bar{\boldsymbol{x}})} |\bar{x}_i|^p = 0$, and $\nabla h(\bar{\boldsymbol{x}}) = \boldsymbol{v}$. Then, for arbitrarily chosen $j \notin I(\bar{\boldsymbol{x}})$, $h(\bar{\boldsymbol{x}} + s\boldsymbol{e}^j) \leq \lambda \sum_{i \in I(\bar{\boldsymbol{x}})} |\bar{x}_i|^p = 0$ for any $s \in \mathbb{R}$ sufficiently small. From this fact along with $h(\bar{\boldsymbol{x}}) = 0$, we see that $s = 0$ is a local maximizer of $\max_{s \in \mathbb{R}} h(\bar{\boldsymbol{x}} + s\boldsymbol{e}^j)$, and thus $v_j = \partial h(\bar{\boldsymbol{x}})/\partial x_j = \partial h(\bar{\boldsymbol{x}} + s\boldsymbol{e}^j)/\partial s|_{s=0} = 0$. Hence, since the index $j \in I(\bar{\boldsymbol{x}})$ was arbitrarily chosen, we obtain the converse implication for (A.12). Using this fact and (A.12), we have

$$\hat{\partial} \hat{g}(\bar{\boldsymbol{x}}) = \{\boldsymbol{v} \mid v_i = 0 \ (i \notin I(\bar{\boldsymbol{x}}))\}. \tag{A.13}$$

We next prove that

$$\partial \hat{g}(\bar{\boldsymbol{x}}) \subseteq \{\boldsymbol{v} \mid v_i = 0 \ (i \notin I(\bar{\boldsymbol{x}}))\}. \tag{A.14}$$

Choose $\boldsymbol{v} \in \partial \hat{g}(\bar{\boldsymbol{x}})$ arbitrarily. Then, there exist sequences $\{\boldsymbol{x}^\nu\}$ and $\{\boldsymbol{v}^\nu\}$ such that $\lim_{\nu \to \infty} \boldsymbol{x}^\nu = \bar{\boldsymbol{x}}$, $\lim_{\nu \to \infty} \boldsymbol{v}^\nu = \boldsymbol{v}$, and $\boldsymbol{v}^\nu \in \hat{\partial} \hat{g}(\boldsymbol{x}^\nu)$ for any ν. For an arbitrary $j \notin I(\bar{\boldsymbol{x}})$, it is not difficult to verify $\boldsymbol{v}_j^\nu = 0$ for all ν sufficiently large. Therefore, we obtain $v_j = 0$ for any $j \notin I(\bar{\boldsymbol{x}})$. Thus, we conclude (A.14) which together with the facts of $\hat{\partial} \hat{g}(\bar{\boldsymbol{x}}) \subseteq \partial \hat{g}(\bar{\boldsymbol{x}})$ and (A.13) implies

$$\partial \hat{g}(\bar{\boldsymbol{x}}) = \{\boldsymbol{v} \mid v_i = 0 \ (i \notin I(\bar{\boldsymbol{x}}))\}.$$

Finally, from this equality and (A.11), we obtain the desired result (A.9).

For the case where $p = 1$, it is easy to show the desired result (A.10) using the fact of $\partial \hat{g}(\bar{\boldsymbol{x}}) = \lambda \sum_{i \in I(\bar{\boldsymbol{x}})} \partial_{\boldsymbol{x}} |x_i| |_{\boldsymbol{x}=\bar{\boldsymbol{x}}}$. We omit the detailed proof. ■

We are now ready to show Lemma 3.

Proof of Lemma 3: We first note that, since G is continuously differentiable and $R_1(\boldsymbol{w}) = \|\boldsymbol{w}\|_p^p$ and $\lambda_1 \geq 0$, we have

$$\begin{aligned}
&\partial_{\boldsymbol{w}} \left(G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 R_1(\boldsymbol{w})\right) \\
=&\nabla_{\boldsymbol{w}} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \partial_{\boldsymbol{w}}(\lambda_1 R_1(\boldsymbol{w})) \\
=&\begin{cases} \left\{\boldsymbol{v} \mid v_i = \frac{\partial G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})}{\partial w_i} + \lambda_1 p \operatorname{sgn}(w_i)|w_i|^{p-1} \ (i \notin I(\boldsymbol{w})), \ v_i \in \mathbb{R} \ (i \in I(\boldsymbol{w}))\right\} & (p < 1) \\ \left\{\boldsymbol{v} \mid v_i = \frac{\partial G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})}{\partial w_i} + \lambda_1 \operatorname{sgn}(w_i) \ (i \notin I(\boldsymbol{w})), v_i \in \frac{\partial G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})}{\partial w_i} + [-\lambda_1, \lambda_1] \ (i \in I(\boldsymbol{w}))\right\} & (p = 1), \end{cases}
\end{aligned} \tag{A.15}$$

where the first equality follows from Proposition A.2 and the second equality comes from Proposition A.4.

Now, let us show the first claim. Suppose $\mathbf{0} \in \partial_{\boldsymbol{w}}(G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 R_1(\boldsymbol{w}))$. Then, by (A.15), we have

$$w_i = 0 \; (i \in I(\boldsymbol{w})), \tag{A.16}$$

$$\frac{\partial G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})}{\partial w_i} + p \operatorname{sgn}(w_i) \lambda_1 |w_i|^{p-1} = 0 \; (i \notin I(\boldsymbol{w})), \tag{A.17}$$

which readily imply $\boldsymbol{W} \nabla_{\boldsymbol{w}} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + p\lambda_1 |\boldsymbol{w}|^p = \mathbf{0}$. Hence, we obtain the first claim.

We next show the latter claim for the case of $p < 1$. Suppose that $\boldsymbol{W} \nabla_{\boldsymbol{w}} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + p\lambda_1 |\boldsymbol{w}|^p = \mathbf{0}$. Then, we see that (A.16) and (A.17) hold. In view of this fact together with (A.15) for $p < 1$, we obtain $\mathbf{0} \in \partial_{\boldsymbol{w}}(G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}}) + \lambda_1 R_1(\boldsymbol{w}))$. Thus, we conclude the latter claim.
■

A.3 Proof of Proposition 6

Denote $\boldsymbol{w}^k = (w_1^k, w_2^k, \ldots, w_n^k)^\top$ for each k. We first show (21). Note that it follows from (19) that

$$w_i^k (\nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_i = p(w_i^k)^2((w_i^k)^2 + \mu_{k-1}^2)^{\frac{p}{2}-1} \tag{A.18}$$

for each $i \in \{1, 2, \ldots, n\}$. Then, for the index $i \notin I(\boldsymbol{w}^*)$, we have $w_i^* \neq 0$ and thus get

$$\begin{aligned} \lim_{k \to \infty} w_i^k (\nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_i &= p(w_i^*)^2 |w_i^*|^{p-2} \\ &= p|w_i^*|^p. \end{aligned} \tag{A.19}$$

We next choose $i \in I(\boldsymbol{w}^*)$ arbitrarily and divide the index set $K := \{1, 2, \ldots, \}$ into the following two sets:

$$U_1^i := \{k \in K \mid w_i^k \neq 0\}, \; U_2^i := \{k \in K \mid w_i^k = 0\}.$$

Then, for $k \in U_1^i$, equation (A.18) together with $p/2 - 1 < 0$ and $w_i^k \neq 0$ yields that

$$\begin{aligned} w_i^k (\nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_i &\leq p|w_i^k|^2 |w_i^k|^{2(\frac{p}{2}-1)} \\ &= p|w_i^k|^p. \end{aligned} \tag{A.20}$$

Since $w_i^k (\nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_i \geq 0$ holds for each $k \in U_1^i$ in view of the right-hand of (A.18) and $\lim_{k \to \infty} \mu_{k-1} = 0$, letting $k \in U_1^i \to \infty$ in (A.20) implies

$$\lim_{k \in U_1^i \to \infty} w_i^k (\nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_i = p|w_i^*|^p = 0. \tag{A.21}$$

Similarly, for all $k \in U_2^i$, we have $w_i^k (\nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_i = 0$ because of $w_i^k = 0 \; (k \in U_2^i)$ and (19). This fact together with (A.21) yields

$$\lim_{k \to \infty} w_i^k (\nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_i = p|w_i^*|^p. \tag{A.22}$$

Combining this with (A.19), we conclude (21).

We next show (22). In view of (20), we have

$$\begin{aligned}
&(w_i^k)^2(\nabla^2\varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_{ii}\\
&= p(w_i^k)^2((w_i^k)^2+\mu_{k-1}^2)^{\frac{p}{2}-1}\\
&\qquad + p(p-2)(w_i^k)^4((w_i^k)^2+\mu_{k-1}^2)^{\frac{p}{2}-2}\\
&= \left(1+\frac{(p-2)(w_i^k)^2}{(w_i^k)^2+\mu_{k-1}^2}\right)\left(w_i^k\left(\nabla\varphi(\boldsymbol{w}^k)\right)_i\right),
\end{aligned} \tag{A.23}$$

for any $i = 1, 2, \ldots, n$, where the last equality is due to (19). For the case of $i \notin I(\boldsymbol{w}^*)$, we obtain

$$\lim_{k\to\infty}\frac{(w_i^k)^2}{(w_i^k)^2+\mu_{k-1}^2} = 1,$$

which together with (A.19) and (A.23) implies

$$\lim_{k\to\infty}(w_i^k)^2(\nabla^2\varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_{ii} = p(p-1)|w_i^*|^p. \tag{A.24}$$

In turn, let us focus on the case of $i \in I(\boldsymbol{w}^*)$. Then, the sequence $\{(w_i^k)^2/\big((w_i^k)^2+\mu_{k-1}^2\big)\}$ is bounded since $|(w_i^k)^2/\big((w_i^k)^2+\mu_{k-1}^2\big)| < 1$ follows from $\mu_{k-1} > 0$ for all k. Hence, using (A.22), we derive from (A.23) that

$$\lim_{k\to\infty}(w_i^k)^2(\nabla^2\varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_{ii} = 0 = p(p-1)|w_i^*|^p,$$

where the last equality is due to $w_i^* = 0$ for $i \in I(\boldsymbol{w}^*)$. By this equation together with (A.24), we conclude (22). The proof is complete.
■

A.4 Proof of Lemma 7

Choose $i \in I(\boldsymbol{w}^*)$ arbitrarily. We show the claim for the case where $w_i^k \neq 0$ for all $k \in K$. It is not difficult to extend the argument to the general case where $w_i^k = 0$ occurs for infinitely many k. Also, we may assume $\lambda_1^k > 0$ for all $k \in K$ because of Assumption A1. For simplicity, denote

$$F_i(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k) := \frac{\partial G(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k)}{\partial w_i}$$

for each $k \in K$. From (19) and the i-th element of condition (7) with $(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\varepsilon}_4) = (\boldsymbol{w}^k, \boldsymbol{\lambda}^k, \boldsymbol{\varepsilon}_4^{k-1})$, we have, for each $k \in K$,

$$F_i(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k) + p\lambda_1^k w_i^k((w_i^k)^2+\mu_{k-1}^2)^{\frac{p}{2}-1} = (\boldsymbol{\varepsilon}_4^{k-1})_i, \tag{A.25}$$

which together with the assumption $w_i^k \neq 0$ and $\lambda_1^k \neq 0$ $(k \in K)$ implies

$$F_i(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k) - (\boldsymbol{\varepsilon}_4^{k-1})_i \neq 0 \ (k \in K).$$

Recall that $\boldsymbol{\varepsilon}_4^{k-1} \to \mathbf{0}$ as $k \to \infty$. Noting this fact and (A.25), we get

$$\mu_{k-1}^2 = \frac{\left|F_i(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k) - (\boldsymbol{\varepsilon}_4^{k-1})_i\right|^{\frac{2}{p-2}}}{\tilde{p}\tilde{\lambda}_1^k|w_i^k|^{\frac{2}{p-2}}} - (w_i^k)^2,$$

where

$$\tilde{p} := p^{\frac{2}{p-2}},\ \tilde{\lambda}_1^k := (\lambda_1^k)^{\frac{2}{p-2}}.$$

Then, it follows that

$$\frac{|w_i^k|^{\frac{2}{2-p}}}{\mu_{k-1}^2} = \frac{\tilde{p}\tilde{\lambda}_1^k}{\left|F_i(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k) - (\varepsilon_4^{k-1})_i\right|^{\frac{2}{p-2}} - \tilde{p}\tilde{\lambda}_1^k |w_i^k|^{2+\frac{2}{p-2}}}. \tag{A.26}$$

To show the desired result, it suffices to prove that $\left\{|w_i^k|^{\frac{2}{2-p}}/\mu_{k-1}^2\right\}_{k\in K}$ is bounded from above. To this end, we first consider the case of $p = 1$. By substituting $p = 1$ for (A.25), we get

$$F_i(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k) - (\varepsilon_4^{k-1})_i + \lambda_1^k \frac{w_i^k}{\sqrt{(w_i^k)^2 + \mu_{k-1}^2}} = 0. \tag{A.27}$$

Moreover, by substituting $p = 1$ for (A.26), we have

$$\begin{aligned} \frac{|w_i^k|^2}{\mu_{k-1}^2} &= \frac{(\lambda_1^k)^{-2}}{\left|F_i(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k) - (\boldsymbol{\varepsilon}_4^{k-1})_i\right|^{-2} - (\lambda_1^k)^{-2}} \\ &= \frac{\left|F_i(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k) - (\boldsymbol{\varepsilon}_4^{k-1})_i\right|^2}{(\lambda_1^k)^2 - \left|F_i(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k) - (\boldsymbol{\varepsilon}_4^{k-1})_i\right|^2}. \end{aligned} \tag{A.28}$$

From equation (A.27), it is not difficult to see that $|F_i(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k) - (\varepsilon_4^{k-1})_i|^2 \leq |\lambda_1^k|^2$. In this inequality, let $k \in K \to \infty$. Then, Assumption A3 together with $F_i(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*) = \partial G(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*)/\partial w_i$ yields

$$(\lambda_1^*)^2 - |F_i(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*)|^2 > 0. \tag{A.29}$$

Letting $k \in K \to \infty$ in equation (A.28) and noting (A.29), we readily derive that

$$\lim_{k\in K\to\infty} \frac{|w_i^k|^{\frac{2}{2-p}}}{\mu_{k-1}^2} = \frac{\left|F_i(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*)\right|^2}{(\lambda_1^*)^2 - \left|F_i(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*)\right|^2} < \infty. \tag{A.30}$$

We next consider the case of $p < 1$. By using (A.26) again, it holds that

$$\begin{aligned} \lim_{k\in K\to\infty} \frac{|w_i^k|^{\frac{2}{2-p}}}{\mu_{k-1}^2} &= \frac{\tilde{p}\tilde{\lambda}_1^*}{\left|F_i(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*)\right|^{\frac{2}{p-2}} - \tilde{p}\tilde{\lambda}_1^* |w_i^*|^{2+\frac{2}{p-2}}} \\ &= \frac{\tilde{p}\tilde{\lambda}_1^*}{\left|F_i(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*)\right|^{\frac{2}{p-2}}} \\ &< \infty, \end{aligned} \tag{A.31}$$

where $\tilde{\lambda}_1^* := (\lambda_1^*)^{\frac{2}{p-2}} > 0$ and the second equality follows from $2 + \frac{2}{p-2} > 0$ and $w_i^* = 0$ because of $i \in I(\boldsymbol{w}^*)$. Particularly, note that the last strict inequality is true due to $2/(p-2) < 0$ even if $\left|F_i(\boldsymbol{w}^*, \bar{\boldsymbol{\lambda}}^*)\right| = 0$. Finally, by (A.30) and (A.31), we conclude the desired result.
■

A.5 Proof of Proposition 9

We prepare the following lemma.

Lemma A.5 *Suppose that Assumption A4 holds and let* $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ *be an arbitrary accumulation point of the sequence* $\{(\boldsymbol{w}^k, \boldsymbol{\lambda}^k)\}$*. Recall that we write* $\nabla_{\tilde{\boldsymbol{w}}} h(\boldsymbol{w}) := \left(\frac{\partial h(\boldsymbol{w})}{\partial w_{i_1}}, \dots, \frac{\partial h(\boldsymbol{w})}{\partial w_{i_p}}\right)^\top \in \mathbb{R}^p$ *for a function* $h : \mathbb{R}^n \to \mathbb{R}$ *and the index set* $\{i_1, i_2, \dots, i_p\} := \{1, 2, \dots, n\} \setminus I(\boldsymbol{w}^*)$*. Moreover, denote* $\tilde{\boldsymbol{w}} := (w_i)_{i \notin I(w^*)}$ *and*

$$\nabla_{(\tilde{\boldsymbol{w}}, \boldsymbol{\lambda})} \Phi_i(\boldsymbol{w}, \boldsymbol{\lambda}) := \begin{bmatrix} \nabla_{\tilde{\boldsymbol{w}}} \Phi_i(\boldsymbol{w}, \boldsymbol{\lambda}) \\ \nabla_{\boldsymbol{\lambda}} \Phi_i(\boldsymbol{w}, \boldsymbol{\lambda}) \end{bmatrix} \in \mathbb{R}^{n - |I(\boldsymbol{w}^*)| + r} \ \ (i \notin I(\boldsymbol{w}^*)), \tag{A.32}$$

$$\nabla_{(\tilde{\boldsymbol{w}}, \boldsymbol{\lambda})} \lambda_i := \begin{bmatrix} \nabla_{\tilde{\boldsymbol{w}}} \lambda_i \\ \nabla_{\boldsymbol{\lambda}} \lambda_i \end{bmatrix} \in \mathbb{R}^{n - |I(\boldsymbol{w}^*)| + r} \ \ (i \in I(\boldsymbol{\lambda}^*)). \tag{A.33}$$

Then, the vectors

$$\left\{ \left\{ \nabla_{(\tilde{\boldsymbol{w}}, \boldsymbol{\lambda})} \Phi_i(\boldsymbol{w}^*, \boldsymbol{\lambda}^*) \right\}_{i \notin I(\boldsymbol{w}^*)}, \left\{ \nabla_{(\tilde{\boldsymbol{w}}, \boldsymbol{\lambda})} \lambda_i |_{\boldsymbol{\lambda} = \boldsymbol{\lambda}^*} \right\}_{i \in I(\boldsymbol{\lambda}^*)} \right\}$$

are linearly independent.

Proof Notice that $\nabla_{(\tilde{\boldsymbol{w}}, \boldsymbol{\lambda})} \lambda_i$ is the vector such that the $(n - |I(\boldsymbol{w}^*)| + i)$-th entry is 1 and the others are 0s. Under Assumption A4, we see that the matrix

$$\boldsymbol{M} := \left[(\nabla \Phi_i(\boldsymbol{w}^*, \boldsymbol{\lambda}^*))_{i \notin I(\boldsymbol{w}^*)}, (\nabla_{(\boldsymbol{w}, \boldsymbol{\lambda})} w_i |_{\boldsymbol{w} = \boldsymbol{w}^*})_{i \in I(\boldsymbol{w}^*)}, (\nabla_{(\boldsymbol{w}, \boldsymbol{\lambda})} \lambda_i |_{\boldsymbol{\lambda} = \boldsymbol{\lambda}^*})_{i \in I(\boldsymbol{\lambda}^*)} \right] \in \mathbb{R}^{(n+r) \times (n + |I(\boldsymbol{\lambda}^*)|)}$$

is of full-column rank. Since the matrix

$$\begin{aligned} \boldsymbol{N} := &\begin{bmatrix} \mathrm{zeros}(|I(\boldsymbol{w}^*)|, n - |I(\boldsymbol{w}^*)|) & \boldsymbol{E}_{|I(\boldsymbol{w}^*)|} & \mathrm{zeros}(|I(\boldsymbol{w}^*)|, |I(\boldsymbol{\lambda}^*)|) \\ \left(\nabla_{(\tilde{\boldsymbol{w}}, \boldsymbol{\lambda})} \Phi_i(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)\right)_{i \notin I(\boldsymbol{w}^*)} & \mathrm{zeros}(n - |I(\boldsymbol{w}^*)| + r, |I(\boldsymbol{w}^*)|) & (\nabla_{(\tilde{\boldsymbol{w}}, \boldsymbol{\lambda})} \lambda_i |_{\boldsymbol{\lambda} = \boldsymbol{\lambda}^*})_{i \in I(\boldsymbol{\lambda}^*)} \end{bmatrix} \\ &\in \mathbb{R}^{(n+r) \times (n + |I(\boldsymbol{\lambda}^*)|)}, \end{aligned}$$

where $\boldsymbol{E}_s$ denotes the $s \times s$ identity matrix and $\mathrm{zeros}(s, t)$ stands for the zero matrix in $\mathbb{R}^{s \times t}$, is obtained by applying appropriate elementary column and row operations to $\boldsymbol{M}$, we find that $\boldsymbol{N}$ is of full-column rank. Hence, the desired result is obtained. ■

Proof of Proposition 9: For simplicity, let

$$\boldsymbol{\xi}^k := ((\boldsymbol{\zeta}^k)^\top, (\boldsymbol{\eta}^k)^\top)^\top, \ \hat{\boldsymbol{\zeta}}^k := \frac{\boldsymbol{\zeta}^k}{\|\boldsymbol{\xi}^k\|}, \ \hat{\boldsymbol{\eta}}^k := \frac{\boldsymbol{\eta}^k}{\|\boldsymbol{\xi}^k\|}$$

for each k. Suppose to the contrary that $\{\boldsymbol{\xi}^k\}$ is unbounded. Choosing an arbitrary accumulation point $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ of the sequence $\{(\boldsymbol{w}^k, \boldsymbol{\lambda}^k)\}$, without loss of generality, we can assume that $(\boldsymbol{w}^k, \boldsymbol{\lambda}^k) \to (\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ and $\|\boldsymbol{\xi}^k\| \to \infty$ as $k \to \infty$, if necessary, by taking a subsequence. Let us denote an arbitrary accumulation point of $\{\boldsymbol{\xi}^k / \|\boldsymbol{\xi}^k\|\}$ by $\hat{\boldsymbol{\xi}}^* := ((\hat{\boldsymbol{\zeta}}^*)^\top, \hat{\boldsymbol{\eta}}^*)^\top$, where $\hat{\boldsymbol{\zeta}}^*$ and $\hat{\boldsymbol{\eta}}^*$ are accumulation points of $\{\hat{\boldsymbol{\zeta}}^k\}$ and $\{\hat{\boldsymbol{\eta}}^k\}$, respectively. Again, without loss of generality, we can suppose $\lim_{k \to \infty} \hat{\boldsymbol{\xi}}^k = \hat{\boldsymbol{\xi}}^*$.

Notice that $\|\hat{\boldsymbol{\xi}}^*\| = 1$. By dividing both sides of (4), (5), (6), and (8) with $\boldsymbol{w} = \boldsymbol{w}^k, \boldsymbol{\lambda} = \boldsymbol{\lambda}^k, \boldsymbol{\zeta} = \boldsymbol{\zeta}^k, \boldsymbol{\eta} = \boldsymbol{\eta}^k$ and $(\boldsymbol{\varepsilon}_1, \varepsilon_2, \boldsymbol{\varepsilon}_3, \boldsymbol{\varepsilon}_4, \varepsilon_5) = (\boldsymbol{\varepsilon}_1^{k-1}, \varepsilon_2^{k-1}, \boldsymbol{\varepsilon}_3^{k-1}, \boldsymbol{\varepsilon}_4^{k-1}, \varepsilon_5^{k-1})$ by $\|\boldsymbol{\xi}^k\|$, we have, for each k,

$$\frac{\left(\nabla f(\boldsymbol{w}^k)\right)_i}{\|\boldsymbol{\xi}^k\|} + \left(\nabla^2_{\boldsymbol{ww}} G(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k)\hat{\boldsymbol{\zeta}}^k\right)_i + \lambda_1^k (\nabla^2 \varphi_{\mu_k}(\boldsymbol{w}^k))_{ii} \hat{\zeta}_i^k = \frac{(\boldsymbol{\varepsilon}_1^{k-1})_i}{\|\boldsymbol{\xi}^k\|} \ \ (i = 1, 2, \ldots, n), \tag{A.34}$$

$$\nabla \varphi_{\mu_k}(\boldsymbol{w}^k)^\top \hat{\boldsymbol{\zeta}}^k - \hat{\eta}_1^k = \frac{\varepsilon_2^{k-1}}{\|\boldsymbol{\xi}^k\|}, \tag{A.35}$$

$$\nabla R_i(\boldsymbol{w}^k)^\top \hat{\boldsymbol{\zeta}}^k - \hat{\eta}_i^k = \frac{(\boldsymbol{\varepsilon}_3^{k-1})_i}{\|\boldsymbol{\xi}^k\|} \ \ (i = 2, 3, \ldots, r), \tag{A.36}$$

$$\lambda_i^k \hat{\eta}_i^k \leq \frac{\varepsilon_5^{k-1}}{\|\boldsymbol{\xi}^k\|}, \ \lambda_i^k \geq 0, \ \hat{\eta}_i^k \geq 0 \ (i = 1, 2, \ldots, r), \tag{A.37}$$

where the last conditions are deduced by componentwise decomposition of (8). Note that $\boldsymbol{\varepsilon}_1^{k-1}/\|\boldsymbol{\xi}^k\|$, $\varepsilon_2^{k-1}/\|\boldsymbol{\xi}^k\|$, $\boldsymbol{\varepsilon}_3^{k-1}/\|\boldsymbol{\xi}^k\|$, and $\varepsilon_5^{k-1}/\|\boldsymbol{\xi}^k\|$ converge to 0 as $k \to \infty$. By driving $k \to \infty$ in (A.37) for $i = 1$ and using $\lim_{k\to\infty} \lambda_1^k = \lambda_1^* > 0$ from Assumption A1, we have

$$\hat{\eta}_1^* = 0. \tag{A.38}$$

In a similar manner, we can get

$$\hat{\eta}_i^* = 0 \ \ (i \notin I(\boldsymbol{\lambda}^*)), \tag{A.39}$$

where $I(\boldsymbol{\lambda}^*) = \{i \in \{1, 2, \ldots, r\} \mid \lambda_i^* = 0\}$ as is defined in Assumption A4. Expressions (A.38) and (A.39) together with $\|\hat{\boldsymbol{\xi}}^*\| = 1$, that is, $\|\hat{\boldsymbol{\xi}}^*\|^2 = \|\hat{\boldsymbol{\zeta}}^*\|^2 + \sum_{i=1}^r |\hat{\eta}_i^*|^2 = 1$ imply

$$\|\hat{\boldsymbol{\zeta}}^*\|^2 + \sum_{i \in I(\boldsymbol{\lambda}^*)} |\hat{\eta}_i^*|^2 = 1. \tag{A.40}$$

Next, let $k \to \infty$ in (A.34). By the boundedness of $\{\nabla^2_{\boldsymbol{ww}} G(\boldsymbol{w}^k, \bar{\boldsymbol{\lambda}}^k)\hat{\boldsymbol{\zeta}}^k\}$ and $\lim_{k\to\infty} \nabla f(\boldsymbol{w}^k)/\|\boldsymbol{\xi}^k\| = \boldsymbol{0}$, we find that $\left\{\lambda_1^k \left(\nabla^2 \varphi_{\mu_k}(\boldsymbol{w}^k)\right)_{ii} \zeta_i^k/\|\boldsymbol{\xi}^k\|\right\}$ is bounded for each i. Using this fact, $\lim_{k\to\infty} \lambda_1^k = \lambda_1^* > 0$, and $\lim_{k\to\infty} |(\nabla^2 \varphi_{\mu_k}(\boldsymbol{w}^k))_{ii}| \to \infty$ for $i \in I(\boldsymbol{w}^*)$ by Proposition 8 yield

$$\hat{\zeta}_i^* = 0 \ \ (i \in I(\boldsymbol{w}^*)). \tag{A.41}$$

We next show that

$$\sum_{i \notin I(\boldsymbol{w}^*)} \operatorname{sgn}(w_i^*)|w_i^*|^{p-1}\hat{\zeta}_i^* = 0. \tag{A.42}$$

For proving (A.42), it suffices to show

$$\lim_{k\to\infty} \nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k)^\top \hat{\boldsymbol{\zeta}}^k = \sum_{i \notin I(\boldsymbol{w}^*)} p \operatorname{sgn}(w_i^*)|w_i^*|^{p-1}\hat{\zeta}_i^*. \tag{A.43}$$

Indeed, we can derive (A.42) from (A.43) by taking the limit of (A.35), (A.41), and (A.38) into account. Choose $i \in I(\boldsymbol{w}^*)$ arbitrarily. By Lemma 7, there exists some $\gamma > 0$ such that

$$\mu_{k-1}^2 \geq \gamma |w_i^k|^{\frac{2}{2-p}} \tag{A.44}$$

for all k sufficiently large. In what follows, we consider sufficiently large k so that inequality (A.44) holds. Then, by $0 < p \leq 1$, we get

$$\frac{\mu_{k-1}^{2-p}}{\gamma^{\frac{2-p}{2}}} \geq |w_i^k|,$$

which implies

$$\begin{aligned}
\frac{1}{p}(\nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_i \hat{\zeta}_i^k =& \left| w_i^k ((w_i^k)^2 + \mu_{k-1}^2)^{\frac{p}{2}-1} \hat{\zeta}_i^k \right| \\
\leq& \left| w_i^k \mu_{k-1}^{2(\frac{p}{2}-1)} \hat{\zeta}_i^k \right| \\
\leq& \frac{\mu_{k-1}^{2-p}}{\gamma^{\frac{2-p}{2}}} \mu_{k-1}^{2(\frac{p}{2}-1)} \left| \hat{\zeta}_i^k \right| \\
=& \gamma^{\frac{p}{2}-1} \left| \hat{\zeta}_i^k \right|. \qquad \text{(A.45)}
\end{aligned}$$

From relation (A.45) and expression (A.41) we obtain $\lim_{k\to\infty}(\nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_i \hat{\zeta}_i^k = 0$. Since $i \in I(\boldsymbol{w}^*)$ was arbitrarily chosen, it holds that

$$\lim_{k\to\infty} \sum_{i \in I(\boldsymbol{w}^*)} (\nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_i \hat{\zeta}_i^k = 0. \qquad \text{(A.46)}$$

It then follows that

$$\begin{aligned}
\lim_{k\to\infty} \nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k)^\top \hat{\boldsymbol{\zeta}}^k &= \lim_{k\to\infty} \left(\sum_{i \in I(\boldsymbol{w}^*)} \left(\nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k) \right)_i \hat{\zeta}_i^k + \sum_{i \notin I(\boldsymbol{w}^*)} \left(\nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k) \right)_i \hat{\zeta}_i^k \right) \\
&= \lim_{k\to\infty} \sum_{i \notin I(\boldsymbol{w}^*)} \left(\nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k) \right)_i \hat{\zeta}_i^k \\
&= \sum_{i \notin I(\boldsymbol{w}^*)} p \operatorname{sgn}(w_i^*) |w_i^*|^{p-1} \hat{\zeta}_i^*,
\end{aligned}$$

where the second equality follows from (A.46) and the last equality is due to the relation

$$\lim_{k\to\infty} (\nabla \varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_i = p \operatorname{sgn}(w_i^*) |w_i^*|^{p-1} \ \ (i \notin I(\boldsymbol{w}^*)), \qquad \text{(A.47)}$$

which can be derived from (19). Therefore, we conclude the desired expression (A.43) and thus (A.42). In addition to (A.47), for $i \notin I(\boldsymbol{w}^*)$, we obtain from (20) that

$$\lim_{k\to\infty} (\nabla^2 \varphi_{\mu_{k-1}}(\boldsymbol{w}^k))_{ii} = p(p-1)|w_i^*|^{p-2}.$$

Then, forcing $k \to \infty$ in (A.34) yields

$$\left. \frac{\partial \left(\nabla_{\boldsymbol{w}} G(\boldsymbol{w}, \bar{\boldsymbol{\lambda}})^\top \hat{\boldsymbol{\zeta}}^* \right)}{\partial w_i} \right|_{(\boldsymbol{w},\boldsymbol{\lambda})=(\boldsymbol{w}^*,\boldsymbol{\lambda}^*)} + \lambda_1^* p(p-1)|w_i^*|^{p-2} \hat{\zeta}_i^* = 0 \ \ (i \notin I(\boldsymbol{w}^*)),$$

which can be transformed by using (A.41) into

$$\left.\frac{\partial\left(\sum_{j\notin I(\boldsymbol{w}^*)}\frac{\partial G(\boldsymbol{w},\bar{\boldsymbol{\lambda}})}{\partial w_j}\hat{\zeta}_j^*\right)}{\partial w_i}\right|_{(\boldsymbol{w},\boldsymbol{\lambda})=(\boldsymbol{w}^*,\boldsymbol{\lambda}^*)} + \lambda_1^* p\left.\frac{\partial\left(\sum_{j\notin I(\boldsymbol{w}^*)}\operatorname{sgn}(w_j)|w_j|^{p-1}\hat{\zeta}_j^*\right)}{\partial w_i}\right|_{\boldsymbol{w}=\boldsymbol{w}^*}$$
$$= 0 \ \ (i \notin I(\boldsymbol{w}^*)).$$

Put $\tilde{\boldsymbol{w}} := (w_i)_{i\notin I(\boldsymbol{w}^*)}$. Letting $k \to \infty$ in (A.36), we get $\nabla R_i(\boldsymbol{w}^*)^\top\hat{\boldsymbol{\zeta}}^* - \hat{\eta}_i^* = 0 \ \ (i = 2,\dots,r)$, which together with (A.41) implies

$$\sum_{j\notin I(\boldsymbol{w}^*)}\frac{\partial R_i(\boldsymbol{w}^*)}{\partial w_j}\hat{\zeta}_j^* - \hat{\eta}_i^* = 0 \ \ (i = 2,\dots,r). \tag{A.48}$$

Now, let $\boldsymbol{\Psi}^* := (\Psi_i^*)_{i\notin I(\boldsymbol{w}^*)}^\top \in \mathbb{R}^{n-|I(\boldsymbol{w}^*)|}$ with

$$\Psi_i^* := \left.\frac{\partial\left(\sum_{j\notin I(\boldsymbol{w}^*)}\frac{\partial G(\boldsymbol{w},\bar{\boldsymbol{\lambda}})}{\partial w_j}\hat{\zeta}_j^*\right)}{\partial w_i}\right|_{(\boldsymbol{w},\boldsymbol{\lambda})=(\boldsymbol{w}^*,\boldsymbol{\lambda}^*)} + \lambda_1^* p\left.\frac{\partial\left(\sum_{j\notin I(\boldsymbol{w}^*)}\operatorname{sgn}(w_j)|w_j|^{p-1}\hat{\zeta}_j^*\right)}{\partial w_i}\right|_{\boldsymbol{w}=\boldsymbol{w}^*} \tag{A.49}$$

and $\boldsymbol{e}^j \in \mathbb{R}^r$ be the vector such that the j-th element is 1 and others are 0s. In addition, Φ_i $(i \notin I(\boldsymbol{w}^*))$, $\nabla_{(\tilde{\boldsymbol{w}},\boldsymbol{\lambda})}\Phi_i$ $(i \notin I(\boldsymbol{w}^*))$, and $\nabla_{(\tilde{\boldsymbol{w}},\boldsymbol{\lambda})}\lambda_i$ $(i \in I(\boldsymbol{\lambda}^*))$ are the functions defined in Assumption A4, (A.32), and (A.33) in Lemma A.5, respectively. Then, it follows that

$$\begin{aligned}
&\sum_{j\notin I(\boldsymbol{w}^*)}\nabla_{(\tilde{\boldsymbol{w}},\boldsymbol{\lambda})}\Phi_j(\boldsymbol{w}^*,\boldsymbol{\lambda}^*)\hat{\zeta}_j^* - \sum_{j\in I(\boldsymbol{\lambda}^*)}\nabla_{(\tilde{\boldsymbol{w}},\boldsymbol{\lambda})}\lambda_j|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}^*}\hat{\eta}_j^* \\
&= \sum_{j\notin I(\boldsymbol{w}^*)}\begin{bmatrix}\nabla_{\tilde{\boldsymbol{w}}}\Phi_j(\boldsymbol{w}^*,\boldsymbol{\lambda}^*)\\ \nabla_{\boldsymbol{\lambda}}\Phi_j(\boldsymbol{w}^*,\boldsymbol{\lambda}^*)\end{bmatrix}\hat{\zeta}_j^* - \sum_{j\in I(\boldsymbol{\lambda}^*)}\begin{bmatrix}\operatorname{zeros}(n-|I(\boldsymbol{w}^*)|,1)\\ \hat{\eta}_j^*\boldsymbol{e}^j\end{bmatrix} \\
&= \begin{bmatrix}\left(\left.\frac{\partial\left(\sum_{j\notin I(\boldsymbol{\lambda}^*)}\Phi_j(\boldsymbol{w},\boldsymbol{\lambda})\hat{\zeta}_j^*\right)}{\partial w_i}\right|_{(\boldsymbol{w},\boldsymbol{\lambda})=(\boldsymbol{w}^*,\boldsymbol{\lambda}^*)}\right)_{i\notin I(\boldsymbol{w}^*)}^\top \\ -\hat{\boldsymbol{\eta}}^* + \sum_{j\notin I(\boldsymbol{w}^*)}\nabla_{\boldsymbol{\lambda}}\Phi_j(\boldsymbol{w}^*,\boldsymbol{\lambda}^*)\hat{\zeta}_j^*\end{bmatrix} \\
&= \begin{bmatrix}\boldsymbol{\Psi}^* \\ \sum_{j\notin I(\boldsymbol{w}^*)}\hat{\zeta}_j^*\left(p\operatorname{sgn}(w_j^*)|w_j^*|^{p-1}\right) \\ \sum_{j\notin I(\boldsymbol{w}^*)}\frac{\partial R_2(\boldsymbol{w}^*)}{\partial w_j}\hat{\zeta}_j^* - \hat{\eta}_2^* \\ \vdots \\ \sum_{j\notin I(\boldsymbol{w}^*)}\frac{\partial R_r(\boldsymbol{w}^*)}{\partial w_j}\hat{\zeta}_j^* - \hat{\eta}_r^*\end{bmatrix} \\
&= \boldsymbol{0},
\end{aligned} \tag{A.50}$$

where $\operatorname{zeros}(n-|I(\boldsymbol{w}^*)|,1)$ denotes the zero matrix in $\mathbb{R}^{n-|I(\boldsymbol{w}^*)|}$, the second equality follows from (A.39), the third one is from (A.38), definition (A.49) of $\boldsymbol{\Psi}^*$, and easy calculation, and the last one is derived from (A.42), (A.48), and (A.49). Expression (A.50) together with Lemma A.5 entails $\hat{\zeta}_i^* = 0$ $(i \notin I(\boldsymbol{w}^*))$ and $\hat{\eta}_i^* = 0$ $(i \in I(\boldsymbol{\lambda}^*))$. Hence, by (A.41), we obtain $\|\hat{\boldsymbol{\zeta}}^*\|^2 + \sum_{i\in I(\boldsymbol{\lambda}^*)}|\hat{\eta}_i^*|^2 = \boldsymbol{0}$. However, it contradicts (A.40). Therefore, the sequence $\{(\boldsymbol{\zeta}^k,\boldsymbol{\eta}^k)\}$ is bounded.

■

Appendix B. Description of the algorithm for solving the smoothed problem (3)

In this section, we explain the algorithm used for solving problem (3) in the numerical experiment.

B.1 Implicit function based method

In this section, we describe the algorithm that is used for solving the following problem arising by smoothing problems (33) and (34) in the numerical experiments in Section 5:

$$\begin{aligned} \min_{(\boldsymbol{w},\boldsymbol{\lambda})\in\mathbb{R}^n\times\mathbb{R}^{n+1}} \quad & f_{\mathrm{val}}(\boldsymbol{w}) := \|\boldsymbol{A}_{\mathrm{val}}\boldsymbol{w} - \boldsymbol{b}_{\mathrm{val}}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{w} \in \operatorname*{argmin}_{\hat{\boldsymbol{w}}} \left\{ \phi_\mu(\hat{\boldsymbol{w}}, \boldsymbol{\lambda}) := \|\boldsymbol{A}_{\mathrm{tr}}\hat{\boldsymbol{w}} - \boldsymbol{b}_{\mathrm{tr}}\|^2 + e^{\lambda_1}\sum_{i=1}^n (\hat{w}_i^2 + \mu^2)^{\frac{p}{2}} + \nu\hat{\boldsymbol{w}}^\top \boldsymbol{C}(\bar{\boldsymbol{\lambda}})\hat{\boldsymbol{w}} \right\}, \end{aligned} \tag{B.1}$$

where $\nu \in \{0,1\}$ and $\boldsymbol{C}(\bar{\boldsymbol{\lambda}}) := \mathrm{Diag}(\exp(\lambda_i))_{i=2}^{n+1}$. The above problems with $\nu = 0$ and 1 correspond to problems (33) and (34), respectively. Our goal is to compute a KKT triplet $(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\eta}) \in \mathbb{R}^n \times \mathbb{R}^{n+1} \times \mathbb{R}^n$ of the above problem with the constraint replaced by the equality constraint $\nabla_{\boldsymbol{w}}\phi_\mu(\boldsymbol{w}, \boldsymbol{\lambda}) = \boldsymbol{0}$. Namely, we compute $(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\eta})$ which satisfies

$$\Theta(\boldsymbol{w}, \boldsymbol{\eta}) := \begin{bmatrix} \nabla f_{\mathrm{val}}(\boldsymbol{w}) \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \nabla^2_{\boldsymbol{ww}}\phi_\mu(\boldsymbol{w}, \boldsymbol{\lambda}) \\ \nabla^2_{\boldsymbol{w\lambda}}\phi_\mu(\boldsymbol{w}, \boldsymbol{\lambda}) \end{bmatrix} \boldsymbol{\eta} = \boldsymbol{0},\ \nabla_{\boldsymbol{w}}\phi_\mu(\boldsymbol{w}, \boldsymbol{\lambda}) = \boldsymbol{0}, \tag{B.2}$$

where $\nabla^2_{\boldsymbol{w\lambda}}\phi_\mu(\boldsymbol{w}, \boldsymbol{\lambda}) = \nabla_{\boldsymbol{\lambda}}\left(\nabla_{\boldsymbol{w}}\phi_\mu(\boldsymbol{w}, \boldsymbol{\lambda})\right) \in \mathbb{R}^{(n+1)\times n}$.

Given $\widetilde{\boldsymbol{\lambda}}$ and μ, let $\widetilde{\boldsymbol{w}}$ be a stationary point of the smoothed lower-level problem $\min_{\boldsymbol{w}} \phi_\mu(\boldsymbol{w}, \boldsymbol{\lambda})$. According to the standard implicit function theorem, if $\nabla^2_{\boldsymbol{ww}}\phi_\mu(\widetilde{\boldsymbol{w}}, \widetilde{\boldsymbol{\lambda}})$ is of full rank, there exist some open neighborhood $U_{\widetilde{\boldsymbol{\lambda}}}$ of $\widetilde{\boldsymbol{\lambda}}$ and a twice continuously differentiable implicit function $\boldsymbol{w}(\cdot) : U_{\widetilde{\boldsymbol{\lambda}}} \to \mathbb{R}^n$ such that

$$\widetilde{\boldsymbol{w}} = \boldsymbol{w}(\widetilde{\boldsymbol{\lambda}}),\ \nabla_{\boldsymbol{w}}\phi_\mu(\boldsymbol{w}(\boldsymbol{\lambda}), \boldsymbol{\lambda}) = \boldsymbol{0}\ \ (\boldsymbol{\lambda} \in U_{\widetilde{\boldsymbol{\lambda}}}).$$

In $U_{\widetilde{\boldsymbol{\lambda}}}$, we may regard problem (B.1) with the constraint replaced by $\nabla_{\boldsymbol{w}}\phi_\mu(\boldsymbol{w}, \boldsymbol{\lambda}) = \boldsymbol{0}$ as

$$\min_{\boldsymbol{\lambda}\in U_{\widetilde{\boldsymbol{\lambda}}}} \left\{ F(\boldsymbol{\lambda}) := \|\boldsymbol{A}_{\mathrm{val}}\boldsymbol{w}(\boldsymbol{\lambda}) - \boldsymbol{b}_{\mathrm{val}}\|_2^2 \right\}. \tag{B.3}$$

By the implicit function theorem again, we then have

$$\nabla\boldsymbol{w}(\widetilde{\boldsymbol{\lambda}}) = -\nabla^2_{\boldsymbol{w\lambda}}\phi_\mu(\boldsymbol{w}(\widetilde{\boldsymbol{\lambda}}), \widetilde{\boldsymbol{\lambda}})\left(\nabla^2_{\boldsymbol{ww}}\phi_\mu(\boldsymbol{w}(\widetilde{\boldsymbol{\lambda}}), \widetilde{\boldsymbol{\lambda}})\right)^{-1},$$

and hereby the gradient of the objective of problem (B.3) at $\widetilde{\boldsymbol{\lambda}}$ is expressed as follows:

$$\begin{aligned} \nabla F(\widetilde{\boldsymbol{\lambda}}) &= \nabla_{\boldsymbol{\lambda}}\|\boldsymbol{A}_{\mathrm{val}}\boldsymbol{w}(\widetilde{\boldsymbol{\lambda}}) - \boldsymbol{b}_{\mathrm{val}}\|_2^2 \\ &= 2\nabla\boldsymbol{w}(\widetilde{\boldsymbol{\lambda}})\boldsymbol{A}_{\mathrm{val}}^\top\left(\boldsymbol{A}_{\mathrm{val}}\boldsymbol{w}(\widetilde{\boldsymbol{\lambda}}) - \boldsymbol{b}_{\mathrm{val}}\right) \\ &= -2\nabla^2_{\boldsymbol{w\lambda}}\phi_\mu(\boldsymbol{w}(\widetilde{\boldsymbol{\lambda}}), \widetilde{\boldsymbol{\lambda}})\left(\nabla^2_{\boldsymbol{ww}}\phi_\mu(\boldsymbol{w}(\widetilde{\boldsymbol{\lambda}}), \widetilde{\boldsymbol{\lambda}})\right)^{-1}\boldsymbol{A}_{\mathrm{val}}^\top\left(\boldsymbol{A}_{\mathrm{val}}\boldsymbol{w}(\widetilde{\boldsymbol{\lambda}}) - \boldsymbol{b}_{\mathrm{val}}\right). \end{aligned}$$

Algorithm B.1 Implicit function based quasi-Newton method for the smoothed subproblem

```
Require: $\boldsymbol{\lambda}^0 \in \mathbb{R}^{n+1}$, $\epsilon \geq 0$, $\alpha, \beta \in (0,1)$, $\boldsymbol{B}_0 \in S_{++}^{n+1}$ ($S_{++}^{n+1}$: The set of $(n+1) \times (n+1)$ real
    symmetric positive definite matrices); Set $k \leftarrow 0$.
 1: while $\nabla F(\boldsymbol{\lambda}^k) \geq \epsilon$ do
 2:     Find $\boldsymbol{w}^k$ satisfying $\nabla_{\boldsymbol{w}} \phi_\mu(\boldsymbol{w}^k, \boldsymbol{\lambda}^k) = 0$.
 3:     Set $\boldsymbol{d}_{\boldsymbol{\lambda}} \leftarrow -\boldsymbol{B}_k^{-1} \nabla F(\boldsymbol{\lambda}^k)$.
 4:     Find the smallest integer $\ell_k \geq 0$ satisfying
            $F(\boldsymbol{\lambda}^k + \beta^{\ell_k} \boldsymbol{d}_{\boldsymbol{\lambda}}) \leq F(\boldsymbol{\lambda}^k) + \alpha \beta^{\ell_k} \nabla F(\boldsymbol{\lambda}^k)^\top \boldsymbol{d}_{\boldsymbol{\lambda}}$.    (B.4)
        Set $t_k \leftarrow \beta^{\ell_k}$.
 5:     $\boldsymbol{\lambda}^{k+1} \leftarrow \boldsymbol{\lambda}^k + t_k \boldsymbol{d}_{\boldsymbol{\lambda}}$.
 6:     Set $\boldsymbol{B}_{k+1} \in S_{++}^{n+1}$.
 7:     $k \leftarrow k+1$
 8: end while
 9: Set $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{\lambda}}) \leftarrow (\boldsymbol{w}^k, \boldsymbol{\lambda}^k)$.
10: Solve $\Theta(\bar{\boldsymbol{w}}, \bar{\boldsymbol{\lambda}}, \boldsymbol{\eta}) = \mathbf{0}$ for $\boldsymbol{\eta}$ to obtain a Lagrange multiplier $\bar{\boldsymbol{\eta}}$.
output $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\eta}})$
```

By computing the above gradient at each iterate, we can preform the quasi-Newton method (Nocedal and Wright, 2006) for problem (B.3) to have a solution $\boldsymbol{\lambda}^*$ with $\nabla F(\boldsymbol{\lambda}^*) = \mathbf{0}$. Once $\boldsymbol{\lambda}^*$ is gained together with $\boldsymbol{w}(\boldsymbol{\lambda}^*)$, we substitute them into the first equation $\Theta(\boldsymbol{w}, \boldsymbol{\eta}) = \mathbf{0}$ in (B.2) and solve the resultant linear equation $\Theta(\boldsymbol{w}(\boldsymbol{\lambda}^*), \boldsymbol{\eta}) = \mathbf{0}$ for $\boldsymbol{\eta}$ to have a solution, say $\boldsymbol{\eta}^*$. The triplet $(\boldsymbol{w}(\boldsymbol{\lambda}^*), \boldsymbol{\lambda}^*, \boldsymbol{\eta}^*)$ is then nothing but the desired KKT triplet.

The overall algorithm is described as in Algorithm B.1. For the algorithm to work, the full-rankness of $\nabla^2_{\boldsymbol{w}\boldsymbol{w}} \phi_\mu(\widetilde{\boldsymbol{w}}, \widetilde{\boldsymbol{\lambda}})$ is necessary to ensure the existence of the implicit function $\boldsymbol{w}(\cdot)$. This is expected to hold in many instances, although it cannot be guaranteed generally. We must solve the lower-level problem in Line 2 every time ℓ_k is updated while performing linesearch (B.4), and thus how we solve the smoothed lower-level problem affects the overall efficiency of Algorithm B.1. In the subsequent section, we will present a certain Newton-type method for solving the smoothed lower-level problem.

Next, we make a remark on the linesearch procedure in Algorithm B.1. As mentioned previously, we need to solve the smoothed lower-level problem $\min_{\boldsymbol{w}} \phi_\mu(\boldsymbol{w}, \boldsymbol{\lambda}^k + \beta^{\ell_k} \boldsymbol{d}_{\boldsymbol{\lambda}})$ so as to evaluate $F(\boldsymbol{\lambda}^k + \beta^{\ell_k} \boldsymbol{d}_{\boldsymbol{\lambda}})$ every time ℓ_k is incremented. Actually, to compute $F(\boldsymbol{\lambda}^k + \beta^{\ell_k} \boldsymbol{d}_{\boldsymbol{\lambda}})$, we need to know the value of $\boldsymbol{w}(\boldsymbol{\lambda}^k + \beta^{\ell_k} \boldsymbol{d}_{\boldsymbol{\lambda}})$ by solving the equation $\nabla_{\boldsymbol{w}} \phi_\mu(\boldsymbol{w}, \boldsymbol{\lambda}^k + \beta^{\ell_k} \boldsymbol{d}_{\boldsymbol{\lambda}}) = \mathbf{0}$. However, the smoothed lower-level problem $\min_{\boldsymbol{w}} \phi_\mu(\boldsymbol{w}, \boldsymbol{\lambda}^k + \beta^{\ell_k} \boldsymbol{d}_{\boldsymbol{\lambda}})$ is nonconvex when $p < 1$ and thus the set of solutions of $\nabla_{\boldsymbol{w}} \phi_\mu(\boldsymbol{w}, \boldsymbol{\lambda}^k + \beta^{\ell_k} \boldsymbol{d}_{\boldsymbol{\lambda}}) = \mathbf{0}$ is not singleton in general[5]. This fact yields that applying a numerical method to this equation may not return $\boldsymbol{w}(\boldsymbol{\lambda}^k + \beta^{\ell_k} \boldsymbol{d}_{\boldsymbol{\lambda}})$. Nevertheless, in practice, we expect $\boldsymbol{w}(\boldsymbol{\lambda}^k + \beta^{\ell_k} \boldsymbol{d}_{\boldsymbol{\lambda}})$ to be computed successfully by applying, for example, a Newton-type method with $\boldsymbol{w}(\boldsymbol{\lambda}^k)$ as a starting point to the equation, because $\boldsymbol{w}(\boldsymbol{\lambda}^k + \beta^{\ell_k} d_{\boldsymbol{\lambda}})$ actually gets closer to $\boldsymbol{w}(\boldsymbol{\lambda}^k)$ as ℓ_k is increased in the linesearch procedure.

[5] When $p = 1$, $\min_{\boldsymbol{w}} \phi_\mu(\boldsymbol{w}, \boldsymbol{\lambda}^k + \beta^{\ell_k} \boldsymbol{d}_{\boldsymbol{\lambda}})$ is strongly convex minimization in virtue of the term $\sum_{i=1}^n (w_i^2 + \mu^2)^{\frac{p}{2}}$ with $\mu > 0$, and thus its solution set is singleton.

The convergence analysis of Algorithm B.1 can be mostly done in a manner similar to that of the standard quasi-Newton method. Indeed, we can show that any accumulation point of $\{(\boldsymbol{w}^k, \boldsymbol{\lambda}_k)\}$ is a KKT point of the smoothed subproblem under the following two sets of assumptions:

Assumption B.1 *Let $\{\boldsymbol{\lambda}^k\}$ be a sequence produced by Algorithm B.1. Then, the following properties hold:*

1. *The sequence $\{\boldsymbol{\lambda}^k\}$ is bounded.*
2. *There exist some α_1, α_2 $(0 < \alpha_1 \leq \alpha_2)$ such that*

$$\alpha_1 \boldsymbol{E} \preceq \boldsymbol{B}_k \preceq \alpha_2 \boldsymbol{E}$$

for all k, where $\boldsymbol{E}$ is the identity matrix with the same size with $\boldsymbol{B}_k$, and for symmetric matrices $\boldsymbol{X}, \boldsymbol{Y}$, $\boldsymbol{X} \preceq \boldsymbol{Y}$ stands for $\boldsymbol{Y} - \boldsymbol{X}$ is positive semidefinite.

The above assumptions are often made in convergence analysis of the quasi-Newton method, whereas the following assumption is specific to our setting.

Assumption B.2 *$\nabla^2_{\boldsymbol{w}\boldsymbol{w}}\phi_\mu(\boldsymbol{w}^k, \boldsymbol{\lambda}^k)$ is of full rank for each k, and so is $\nabla^2_{\boldsymbol{w}\boldsymbol{w}}\phi_\mu(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$ even at an arbitrary accumulation point $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*)$.*

Assumption B.2 ensures that the implicit function $\boldsymbol{w}(\cdot)$ exists at each iterate and even at an arbitrary accumulation point.

The following theorem holds under Assumptions B.1 and B.2. As the proof is similar to that for the quasi-Newton method, we omit it here.

Theorem B.3 *Suppose that Assumptions B.1 and B.2 hold. Then, any accumulation point of $\{\boldsymbol{\lambda}^k\}$ satisfies $\nabla F(\boldsymbol{\lambda}) = \boldsymbol{0}$.*

B.2 Newton-type method for solving the smoothed lower-level problem

In this section, we describe the modified Newton-type algorithm used for solving the smoothed lower-level problem $\min_{\boldsymbol{w}} \phi_\mu(\boldsymbol{w}, \boldsymbol{\lambda})$ in problem (B.1). For brevity, the algorithm is presented in the form pertaining to the following problem:

$$\min_{\boldsymbol{w}} \ \psi_\mu(\boldsymbol{w}) := \frac{1}{2}\|\boldsymbol{K}\boldsymbol{w} - \boldsymbol{f}\|^2 + \eta \sum_{i=1}^{n} (w_i^2 + \mu^2)^{\frac{p}{2}}, \tag{B.5}$$

where $\eta \in \mathbb{R}$ is positive, $\boldsymbol{K} \in \mathbb{R}^{m \times n}$, and $\boldsymbol{f} \in \mathbb{R}^m$. Note that by setting $\boldsymbol{K}$ and $\boldsymbol{f}$ appropriately, the function ψ_μ above reduces to ϕ_μ.

We begin with the update-formula of the standard Newton method for problem (B.5) at the r-th iterate $\boldsymbol{w}^r \in \mathbb{R}^n$:

$$\boldsymbol{w}^{r+1} \leftarrow \boldsymbol{w}^r - \boldsymbol{B}(\boldsymbol{w}^r)^{-1}\nabla\psi_\mu(\boldsymbol{w}^r), \text{ where}$$

$$\boldsymbol{B}(\boldsymbol{w}) := \boldsymbol{K}^\top\boldsymbol{K} + p\eta\,\mathrm{Diag}\left((w_i^2 + \mu^2)^{\frac{p}{2}-1} + \underbrace{\frac{p-2}{2} w_i^2 (w_i^2 + \mu^2)^{\frac{p}{2}-2}}_{\text{negative}} \right)_{i=1}^{n}.$$

However, the matrix $\boldsymbol{B}(\boldsymbol{w}^r)$ is not necessarily nonsingular because of the above negative part, and thus the Newton method may not work.[6] As a remedy, in the spirit of the modified Newton method, we modify $\boldsymbol{B}(\boldsymbol{w}^r)$ to the following matrix $\widetilde{\boldsymbol{B}}(\boldsymbol{w}^r)$ by deleting the negative part:

$$\widetilde{\boldsymbol{B}}(\boldsymbol{w}) := \boldsymbol{K}^\top \boldsymbol{K} + p\eta \mathrm{Diag}\left((w_i^2 + \mu^2)^{\frac{p}{2}-1}\right)_{i=1}^n.$$

Now, the presented algorithm is described formally as in Algorithm B.2. In fact, the algorithm is identical to the one that is proposed by Lai and Wang (2011, Section 2), which gives the following theorem:

Theorem B.4 *(Lai and Wang, 2011, Theorem 2.1) Let $\{\boldsymbol{w}^r\}$ be a sequence generated by Algorithm B.2 with $\epsilon = 0$. It is bounded and its arbitrary accumulation point satisfies $\nabla\psi_\mu(\boldsymbol{w}) = \boldsymbol{0}$.*

It is worthwhile to note that Algorithm B.2 does not request a linesearch procedure for the global convergence, which is often costly.

Algorithm B.2 Modified Newton-type method for $\min_{\boldsymbol{w}} \psi_\mu(\boldsymbol{w})$

```
Require: $\boldsymbol{w}^0 \in \mathbb{R}^n$, $r \leftarrow 0$, $\epsilon > 0$
while $\|\nabla\psi_\mu(\boldsymbol{w}^r)\| > \epsilon$ do
    $\boldsymbol{w}^{r+1} \leftarrow \boldsymbol{w}^r - \widetilde{\boldsymbol{B}}(\boldsymbol{w}^r)^{-1}\nabla\psi_\mu(\boldsymbol{w}^r)$,
    $r \leftarrow r + 1$.
end while
Set $\bar{\boldsymbol{w}} \leftarrow \boldsymbol{w}^r$
output $\bar{\boldsymbol{w}}$
```

Appendix C. Supplementary tables and figures of `bayesopt` for the numerical experiments

This section provides the supplementary Tables C.1 and C.2 that show the first time when `bayesopt` found the best observed objective value. These results were recorded in a single run of `bayesopt` for each problem, thus differ from the averaged results shown in Tables 1 and 2. In addition, it also gives Figures C.1 and C.2 that depict how the best observed objective value of `bayesopt` varies over time. In order to monitor the change of values in a long period, we extended the time limit of `bayesopt` to 1200 seconds from 600 seconds that was employed for making Tables C.1 and C.2. These figures were obtained by solving the problems organized from the data sets of **CpuSmall** and **Student**.

[6] In fact, when $p = 1$, $\boldsymbol{B}(\boldsymbol{w})$ is nonsingular even in the presence of the negative part, because

$$p\eta\mathrm{Diag}\left((w_i^2+\mu^2)^{\frac{p}{2}-1} + \frac{p-2}{2}w_i^2(w_i^2+\mu^2)^{\frac{p}{2}-2}\right)_{i=1}^n = p\eta\mathrm{Diag}\left(\left(\mu^2 + \frac{p}{2}w_i^2\right)(w_i^2+\mu^2)^{\frac{p}{2}-1}\right)_{i=1}^n,$$

which turns out to be positive definite.

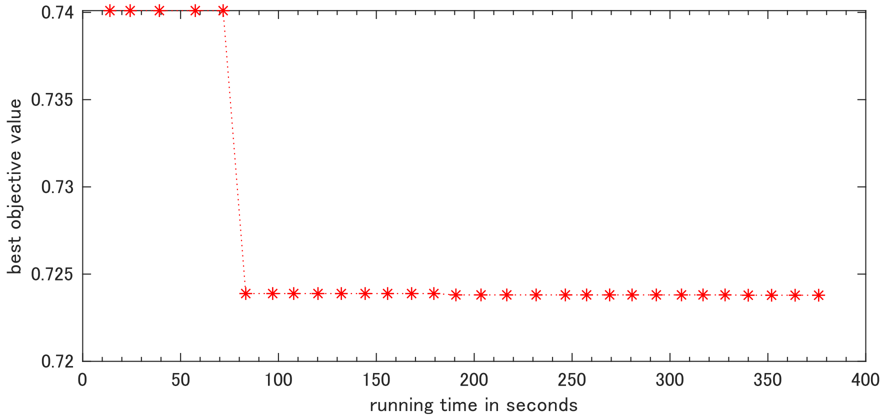
Table C.1: The first time of `bayesopt` for finding a solution of problem (33), which attains the final best observed objective value, that is, validation value (Those results of `bayesopt` were recorded in a single run, thus differ from the averaged results over 5 runs shown in Table 1. For the sake of comparison, the results of Algorithm 1 are also shown, which are the same as those in Table 1. "f.time (sec)" stands for the first time in seconds where the best objective value is observed. The best values in f.time (sec) and $\mathrm{Err}_{\mathrm{val}}$ are displayed in bold.)

Data		`bayesopt`		Algorithm 1	
name	p	$\mathrm{Err}_{\mathrm{val}}$	f.time (sec)	$\mathrm{Err}_{\mathrm{val}}$	time (sec)
Facebook	1	6.476	42.256	**6.474**	**17.399**
	0.8	**6.504**	122.158	6.512	**22.242**
	0.5	**6.536**	66.589	6.550	**16.820**
Insurance	1	**95.764**	49.538	**95.764**	**33.077**
	0.8	95.737	63.580	**95.676**	**32.465**
	0.5	95.604	**13.960**	**95.562**	44.904
Student	1	**0.777**	12.405	0.778	**10.586**
	0.8	**0.724**	339.980	**0.724**	**2.348**
	0.5	0.731	147.118	**0.724**	**3.618**
BodyFat	1	**0.209**	4.839	**0.209**	**0.068**
	0.8	0.180	3.899	**0.179**	**0.203**
	0.5	**0.212**	1.160	0.267	**0.395**
CpuSmall	1	131124	**1.202**	**130981**	11.299
	0.8	131187	1.853	**130982**	**0.741**
	0.5	131234	1.712	**131058**	**0.672**

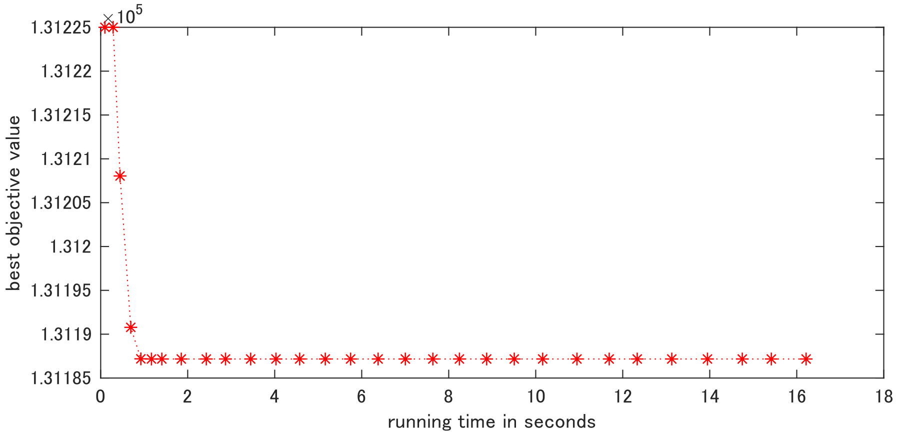
Table C.2: The first time of `bayesopt` for finding a solution of problem (34), which attains the final best observed objective value, that is, validation value (Those results of `bayesopt` were recorded in a single run, thus differ from the averaged results over 5 runs shown in Table 2. For the sake of comparison, the results of Algorithm 1 (Alg.1-A, Alg.1-B) are also shown, which are the same as those in Table 2. "f.time (sec)" stands for the first time in seconds where the best objective value is observed. The best values in f.time (sec) and $\mathrm{Err}_{\mathrm{val}}$ are displayed in bold.)

Data		`bayesopt`		Alg.1-A		Alg.1-B	
name	$\sharp\boldsymbol{\lambda}$	$\mathrm{Err}_{\mathrm{val}}$	f.time (sec)	$\mathrm{Err}_{\mathrm{val}}$	time (sec)	$\mathrm{Err}_{\mathrm{val}}$	time (sec)
Facebook	54	8.780	**1.518**	**6.478**	20.247	–	–
Insurance	86	107.000	**4.288**	95.694	50.714	**94.604**	4.473
Student	273	18.324	26.756	**0.771**	**1.451**	0.786	71.775
BodyFat	15	46.815	0.337	0.243	**0.072**	**0.130**	0.695
CpuSmall	13	151394	531.837	131130	6.222	**128540**	**0.658**

Figure C.1: Best observed objective value (validation value) vs running time in seconds (`bayesopt` for problem (33) with a single $\ell_{0.8}$ hyperparameter)

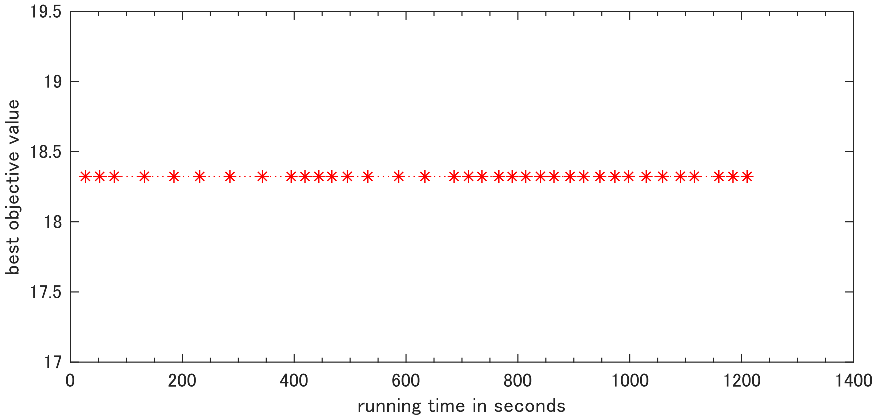


(a) **Student** (The number of hyperparameters is 1; the proposed bilevel algorithm found a solution with 0.724 in 2 seconds.)

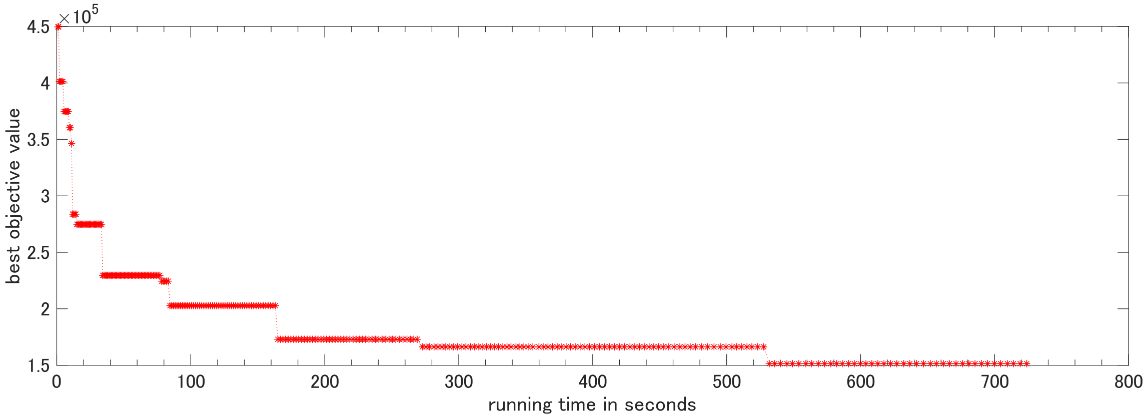


(b) **CpuSmall** (The number of hyperparameters is 1; the proposed bilevel algorithm found a solution with 1.3098×10^5 in 0.7 seconds.)

Figure C.2: Best observed objective value (validation value) vs running time in seconds (`bayesopt` for problem (34) with multiple hyperparameters)



(a) **Student** (The number of hyperparameters is 273; the proposed bilevel algorithms found solutions with 0.771 in 1.45 seconds (Alg.1-A) and 0.786 in 71.78 seconds (Alg.1-B).)



(b) **CpuSmall** (The number of hyperparameters is 13; the proposed bilevel algorithms found solutions with 1.31×10^5 in 6.22 seconds (Alg.1-A) and 1.29×10^5 in 0.66 seconds (Alg.1-B).)

References

Amir Beck and Marc Teboulle. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

Kristin P. Bennett, Jing Hu, Xiaoyun Ji, Gautam Kunapuli, and Jong-Shi Pang. Model selection via bilevel optimization. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 1922–1929, 2006.

Kristin P. Bennett, Gautam Kunapuli, Jing Hu, and Jong-Shi Pang. Bilevel optimization and machine learning. Computational Intelligence: Research Frontiers. WCCI 2008. Lecture Notes in Computer Science, vol. 5050, Berlin, Heidelberg, 2008. Springer.

Wei Bian and Xiaojun Chen. Worst-case complexity of smoothing quadratic regularization methods for non-Lipschitzian optimization. *SIAM Journal on Optimization*, 23(3):1718–1741, 2013.

Wei Bian and Xiaojun Chen. Optimality and complexity for constrained optimization problems with nonconvex regularization. *Mathematics of Operations Research*, 42(4):1063–1084, 2017.

Wei Bian, Xiaojun Chen, and Yinyu Ye. Complexity analysis of interior point algorithms for non-Lipschitz and nonconvex minimization. *Mathematical Programming*, 149(1-2):301–327, 2015.

Emmanuel J. Candes, Michael B. Wakin, and Stephen P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.

Xiaojun Chen. Smoothing methods for nonsmooth, nonconvex minimization. *Mathematical Programming*, 134(1):71–99, 2012.

Xiaojun Chen, Fengmin Xu, and Yinyu Ye. Lower bound theory of nonzero entries in solutions of ℓ_2-ℓ_p minimization. *SIAM Journal on Scientific Computing*, 32(5):2832–2852, 2010.

Xiaojun Chen, Lingfeng Niu, and Yaxiang Yuan. Optimality conditions and a smoothing trust region Newton method for nonLipschitz optimization. *SIAM Journal on Optimization*, 23(3):1528–1552, 2013.

Xiaojun Chen, Dongdong Ge, Zizhuo Wang, and Yinyu Ye. Complexity of unconstrained L_2-L_p minimization. *Mathematical Programming*, 143(1-2):371–383, 2014.

Stephan Dempe, Joydeep Dutta, and Sebastian Lohse. Optimality conditions for bilevel programming problems. *Optimization*, 55(5-6):505–524, 2006.

Stephan Dempe and Alain B. Zemkoho. The generalized Mangasarian-Fromowitz constraint qualification and optimality conditions for bilevel programs. *Journal of Optimization Theory and Applications*, 148(1):46–68, 2011.

Stephan Dempe and Alain B. Zemkoho. The bilevel programming problem: reformulations, constraint qualifications and optimality conditions. *Mathematical Programming*, 138:447–473, 2013.

Stephan Dempe, Vyacheslav Kalashnikov, Gerardo A Pérez-Valdés, and Nataliya Kalashnykova. Bilevel programming problems. *Energy Systems. Springer, Berlin*, 2015.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated Machine Learning*, pages 3–33. Springer, 2019.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1165–1173, 2017.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimilano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1568–1577, 2018.

Dongdong Ge, Xiaoye Jiang, and Yinyu Ye. A note on the complexity of L_p minimization. *Mathematical Programming*, 129(2):285–299, 2011.

Pinghua Gong, Changshui Zhang, Zhaosong Lu, Jianhua Huang, and Jieping Ye. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *Proceedings of the 30th International Conference on Machine Learning*, volume 37, pages 37–45, 2013.

Michael Hintermüller and Tao Wu. Nonconvex TV^q-models in image restoration: Analysis and a trust-region regularization–based superlinearly convergent solver. *SIAM Journal on Imaging Sciences*, 6(3):1385–1415, 2013.

Yaohua Hu, Chong Li, Kaiwen Meng, Jing Qin, and Xiaoqi Yang. Group sparse optimization via $\ell_{p,q}$ regularization. *Journal of Machine Learning Research*, 18:1–52, 2017.

Karl Kunisch and Thomas Pock. A bilevel optimization approach for parameter learning in variational models. *SIAM Journal on Imaging Sciences*, 6(2):938–983, 2013.

Ming-Jun Lai and Jingyue Wang. An unconstrained ℓ_q minimization with $0 < q \leq 1$ for sparse solution of underdetermined linear systems. *SIAM Journal on Optimization*, 21(1):82–101, 2011.

Moshe Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 1540–1552, 2020.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 2113–2122, 2015.

Goran Marjanovic and Victor Solo. On ℓ_q optimization and matrix completion. *IEEE Transactions on Signal Processing*, 60(11):5714–5724, 2012.

Goran Marjanovic and Victor Solo. On exact ℓ_q denoising. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6068–6072, 2013.

Chuang Miao and Hengyong Yu. Alternating iteration for ℓ_p $(0 < p \leq 1)$ regularized CT reconstruction. *IEEE Access*, 4:4355–4363, 2016.

Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of bayesian methods for seeking the extremum. *Towards Global Optimization*, 2:117–129, 1978.

Gregory Moore, Charles Bergeron, and Kristin P. Bennett. Model selection for primal SVM. *Machine Learning*, 85(1):175–208, 2011.

Gregory M Moore, Charles Bergeron, and Kristin P Bennett. Nonsmooth bilevel programming for hyperparameter selection. In *2009 IEEE International Conference on Data Mining Workshops*, pages 374–381, 2009.

Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1): 127–152, 2005.

Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.

Peter Ochs, René Ranftl, Thomas Brox, and Thomas Pock. Techniques for gradient-based bilevel optimization with non-smooth lower level problems. *Journal of Mathematical Imaging and Vision*, 56(2):175–194, 2016.

Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 737–746, 2016.

R. Tyrrell Rockafellar and Roger J-B Wets. *Variational Analysis*, volume 317. Springer Science & Business Media, 2009.

Saharon Rosset. Bi-level path following for cross validated solution of kernel quantile regression. *Journal of Machine Learning Research*, 10:2473–2505, 2009.

Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, volume 89, pages 1723–1732, 2019.

Fei Wen, Peilin Liu, Yipeng Liu, Robert C. Qiu, and Wenxian Yu. Robust sparse recovery for compressive sensing in impulsive noise using ℓ_p-norm model fitting. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4643–4647, 2016.

Fei Wen, Lasith Adhikari, Ling Pei, Roummel F. Marcia, Peilin Liu, and Robert C Qiu. Nonconvex regularization-based sparse recovery and demixing with application to color image inpainting. *IEEE Access*, 5:11513–11527, 2017.

Fei Wen, Lei Chu, Peilin Liu, and Robert C. Qiu. A survey on nonconvex regularization-based sparse and low-rank recovery in signal processing, statistics, and machine learning. *IEEE Access*, 6: 69883–69906, 2018.

Haolei Weng, Le Zheng, Arian Maleki, and Xiaodong Wang. Phase transition and noise sensitivity of ℓ_p-minimization for $0 \leq p \leq 1$. In *IEEE International Symposium on Information Theory*, pages 675–679, 2016.

Zongben Xu, Xiangyu Chang, Fengmin Xu, and Hai Zhang. $L_{1/2}$ regularization: A thresholding representation theory and a fast solver. *IEEE Transactions on neural networks and learning systems*, 23(7):1013–1027, 2012.

Jane J. Ye and D. L. Zhu. Optimality conditions for bilevel programming problems. *Optimization*, 33(1):9–27, 1995.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942, 2010.

Le Zheng, Arian Maleki, Quanhua Liu, Xiaodong Wang, and Xiaopeng Yang. An ℓ_p-based reconstruction algorithm for compressed sensing radar imaging. In *2016 IEEE Radar Conference (RadarConf)*, pages 1–5, 2016.