Lower-level Duality Based Reformulation and Majorization Minimization Algorithm for Hyperparameter Optimization

He Chen[1] **Haochen Xu**[2] **Rujun Jiang**[2†] **Anthony Man-Cho So**[1]
[1]Dept. of SEEM, The Chinese University of Hong Kong
[2]School of Data Science, Fudan University

Abstract

Hyperparameter tuning is an important task of machine learning, which can be formulated as a bilevel program (BLP). However, most existing algorithms are not applicable for BLP with non-smooth lower-level problems. To address this, we propose a single-level reformulation of the BLP based on lower-level duality without involving any implicit value function. To solve the reformulation, we propose a majorization minimization algorithm that marjorizes the constraint in each iteration. Furthermore, we show that the subproblems of the proposed algorithm for several widely-used hyperparameter turning models can be reformulated into conic programs that can be efficiently solved by the off-the-shelf solvers. We theoretically prove the convergence of the proposed algorithm and demonstrate its superiority through numerical experiments.

1 Introduction

Machine learning research is focused on developing methods that can effectively extract important elements from given datasets. Various learning methods has emerged, encompassing biologically inspired neural networks (Bishop et al., 1995), ensemble models (Claesen et al., 2014), adversarial learning (Brückner and Scheffer, 2011; Wang et al., 2021, 2022), and reinforcement learning (Yang et al., 2019; Wu et al., 2020). These methods commonly rely on a set of hyperparameters, which are adjustable parameters that configure various aspects of the learning algorithm. The choice of hyperparameters can significantly impact the resulting model and its performance, leading to a wide range of effects.

Finding the optimal hyperparameters for a machine learning model is often considered one of the most challenging aspects of the workflow. Regularization, a widely employed technique in model fitting for regression and classification tasks, involves adding a regularization penalty to the empirical risk term, thereby controlling complexity. An advanced strategy for adapting hyperparameters is to employ a training/validation approach, which entails optimizing the parameters with regularization on a training set and subsequently evaluating the performance by computing its loss on a separate validation set. Mathematically, the process of hyperparameter selection can be formulated into the following bilevel program (BLP):

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n,\boldsymbol{\lambda}\in\mathbb{R}^\tau_+} \quad & L(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \arg\min_{\hat{\mathbf{x}}\in\mathbb{R}^n} \left\{ l(\hat{\mathbf{x}}) + \sum_{i=1}^{\tau} \lambda_i P_i(\hat{\mathbf{x}}) \right\}, \end{aligned} \tag{1}$$

where $L, l, P_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are proper convex closed functions, $\mathbf{x}$ is the parameter to learn, and $\boldsymbol{\lambda}$ is a vector of hyperparameters. Note that all the functions can be *nonsmooth*. In BLP (1), the upper-level (UL) problem minimizes the validation error affected by the hyperparameters, and the lower-level (LL) problem aims to minimize structural risk on given training data incorporating a regularizer penalized by hyperparameters that need to be tuned. Table Table 1 provides some illustrative examples of bilevel hyperparameter selection problems in form (1).

1.1 Related Work

In the existing literature, various approaches have been proposed for hyperparameter selection. The simpler approaches include brute force grid search and Bayesian optimization, which handle hyperparameters and datasets of small-scale but suffer from high computational requirements. These gradient-free methods face limitations when dealing with a large number of

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. †Corresponding Author.

parameters.

Gradient-based methods for BLPs are popular in literature. It can be broadly categorized into two groups. Explicit Gradient-Based Methods (EGBMs) (Franceschi et al., 2017, 2018) utilize dynamic frameworks and iterative algorithms to solve the LL problem. Implicit Gradient-Based Methods (IGBMs) (Pedregosa, 2016; Rajeswaran et al., 2019; Lorraine et al., 2020) rely on the first-order optimality condition for the LL problem and the chain rule to derive the hyper-gradient by solving a linear system. To mitigate computational complexity, techniques such as the Conjugate Gradient (CG) method or the Neumann method (Pedregosa, 2016; Lorraine et al., 2020) are often employed for fast inverse computations. Recently, Liu et al. (2022) introduce an approach that drops the implicit gradient and Shen and Chen (2023) tackle the bilevel problem through the penalty methods. However, all the above mentioned methods are only applicable to smooth LL problems, and may not be suitable for (1).

For nonsmooth functions, Bertrand et al. (2020) propose a new implicit differentiation method combined with block coordinate descent to solve Lasso-type models for hyperparameter optimization. In their subsequent work (Bertrand et al., 2022), it is extended to solve more general non-smooth hyperparameter optimization problems. However, their methods are restricted to l_1 regularized LL problems, which cannot deal with general P_i. In the context of difference of convex bilevel programs, Ye et al. (2021) develop a numerical algorithm called iP-DCA and applies it to hyperparameter selection, particularly in support vector machine models. Gao et al. (2022) propose the Value Function Based Difference-of-Convex Algorithm (VF-iDCA) to handle BLPs like the one presented in equation (1), where the LL problems involve complex regularization terms. However, these DCA-based methods requires to compute the optimal value of the LL problem to obtain a subgradient, which is used to linearizes the concave term in DC constraint at each iteration. Recently, Chen et al. (2023) have introduced an inexact gradient-free method whose subproblem is a simple bilevel program, which is still difficult to solve.

1.2 Our Motivations and Contributions

This paper presents a novel single-level reformulation for the structured BLP (1). By leveraging Fenchel's duality, the proposed reformulation only requires the expression of the conjugate of each atom function, which is often more easily accessible compared to the value function used in previous works (Bertrand et al., 2022; Gao et al., 2022; Pedregosa, 2016). Moreover, we introduce a Lower-level Duality based Majorization Minimization Algorithm (LDMMA) for hyperparameter selection in form (1). Notably, this algorithm accommodates lower-level problems that are *nonsmooth* and *non-strongly convex* in $\mathbf{x}$. We first reformulate (1) into a single-level problem without involving any value function. However, a drawback of this convex subproblem is that it lacks an interior point, rendering most convex optimization methods ineffective. To remedy this, a small positive constant ϵ is added to the right side of the constraint in (1), leading to a relaxed convex approximation with interior points. Based on this reformulation, we propose an iterative algorithm that sequentially solves convex subproblems using majorization minimization (MM) techniques to approximate the constraint. Furthermore, we show that the subproblems of several widely-used hyperparameter models can be reformulated to conic programs that can be efficiently solved by the off-the-shelf solvers. Additionally, we demonstrate that the obtained solutions converge subsequentially to a Karush-Kuhn-Tucker (KKT) point of the ϵ-perturbation of problem (1) under mild conditions. Numerical experiments are conducted to showcase the efficiency of our method. We summarize our contributions as follows

- We provide a novel reformulation for a class of structured BLPs that is a single-level problem and do not involve value functions.
- Based on the reformulation and using MM techniques, we further propose an iterative algorithm, LDMMA, where the subproblem in each iteration is a convex problem. For many practical applications, the subproblem is a convex conic program.
- Theoretically, we prove that our algorithm generates a sequence whose accumulation points are KKT points under mild conditions.
- We conduct numerical experiments on both synthetic and real-world datasets and show that LDMMA exceeds the state-of-the-art.

2 Fenchel's Duality Based Reformulation for BLP

We propose a Fenchel's duality based reformulation for the following problem, which is a generalization of (1)

$$\min_{\mathbf{x}\in\mathbb{R}^n,\boldsymbol{\lambda}\in\mathcal{D}} f(\mathbf{x},\boldsymbol{\lambda}) \qquad \text{s.t. } \mathbf{x}\in\arg\min_{\hat{\mathbf{x}}} \sum_{i=0}^{\tau} g_i(\hat{\mathbf{x}},\boldsymbol{\lambda}), \quad (2)$$

where $\boldsymbol{\lambda}$ is a vector of hyperparameters in a convex closed set $\mathcal{D}$ and f, g_i are proper convex closed function in $\mathbf{x}$ and $\boldsymbol{\lambda}$ but possibly non-smooth. The core idea is to replace the min operator in LL problem with max operator by invoking Fenchel's duality in the conventional value function reformulation, and then the max

Table 1: Examples of bilevel hyperparameter selection problems of the form (1), see Kunapuli et al. (2008); Feng and Simon (2018) for reference.

Machine learning algorithm	LL variable	UL variable	$L(\mathbf{x})/l(\mathbf{x})$	Regularization
elastic net	$\mathbf{x}$	λ_1, λ_2	$\frac{1}{2}\sum_{i\in I_{val}/i\in I_{tr}} \lvert b_i - \mathbf{x}^T\mathbf{a}_i\rvert^2$	$\lambda_1\lVert\mathbf{x}\rVert_1 + \frac{\lambda_2}{2}\lVert\mathbf{x}\rVert_2^2$
sparse group lasso	$\mathbf{x}$	$\lambda \in \mathbb{R}_+^{M+1}$	$\frac{1}{2}\sum_{i\in I_{val}/i\in I_{tr}} \lvert b_i - \mathbf{x}^T\mathbf{a}_i\rvert^2$	$\sum_{m=1}^M \lambda_m\lVert\mathbf{x}^{(m)}\rVert_2 + \lambda_{M+1}\lVert\mathbf{x}\rVert_1$
support vector machine	$\mathbf{w}, c$	$\lambda, \overline{\mathbf{w}}$	$\sum_{j\in I_{val}/j\in I_{tr}} max(1 - b_j(\mathbf{x}^T\mathbf{a}_j - c), 0)$	$\frac{\lambda}{2}\lVert\mathbf{x}\rVert^2$ (with constraint $-\overline{\mathbf{w}} \le \mathbf{x} \le \overline{\mathbf{w}}$)
low-rank matrix completion	θ, β, Γ	$\lambda \in \mathbb{R}_+^{2G+1}$	$\sum_{(i,j)\in\Omega_{val}/(i,j)\in\Omega_{tr}} \frac{1}{2}\lvert M_{ij} - \mathbf{x}_i\theta - \mathbf{z}_j\beta - \Gamma_{ij}\rvert$	$\lambda_0\lVert\Gamma\rVert_* + \sum_{g=1}^G \lambda_g\lVert\theta^{(g)}\rVert_2 + \sum_{g=1}^G \lambda_{g+G}\lVert\beta^{(g)}\rVert_2$

operator can be omitted due to the direction of the inequality. Hence we obtain an equivalent inequality constraint only involving LL functions and their conjugates. Let us begin with the following equivalent form of LL problem.

$$\min_{\mathbf{x}} g_0(\mathbf{x}, \boldsymbol{\lambda}) + \sum_{i=1}^{\tau} g_i(\mathbf{z}_i, \boldsymbol{\lambda}) \qquad \text{s.t.} \quad \mathbf{x} = \mathbf{z}_i. \tag{3}$$

Since $g_i, i = 0, 1, \ldots, \tau$ are convex and the constraints are affine, it is know that strong duality holds under Slater's condition. That is, if $\cap_{i=0}^{\tau}\text{ri}(\text{dom } g_i(\cdot, \boldsymbol{\lambda})) \neq \emptyset$[1], (3) is equivalent to the following problem:

$$-\min_{\boldsymbol{\rho}_i}\max_{\mathbf{x},\mathbf{z}_i} -g_0(\mathbf{x}, \boldsymbol{\lambda}) - \sum_{i=1}^{\tau} g_i(\mathbf{z}_i, \boldsymbol{\lambda}) - \sum_{i=1}^{\tau} \boldsymbol{\rho}_i^T(\mathbf{x} - \mathbf{z}_i). \tag{4}$$

Here $\boldsymbol{\rho}_i \in \mathbb{R}^n, i = 1, \ldots, \tau$ are Lagrangian multipliers associated with constraint $\mathbf{x} = \mathbf{z}_i$, and the min and max operators have been exchanged by adding the negative signs. We define $g_i^*(\mathbf{y}, \boldsymbol{\lambda}) := \max_{\mathbf{x}}\ \mathbf{y}^T\mathbf{x} - g_i(\mathbf{x}, \boldsymbol{\lambda})$ as the conjugate functions regarding $\mathbf{x}$ for g_i. We then simplify (4) as

$$\max_{\boldsymbol{\rho}_i} -g_0^*\left(-\sum_{i=1}^{\tau}\boldsymbol{\rho}_i, \boldsymbol{\lambda}\right) - \sum_{i=1}^{\tau} g_i^*(\boldsymbol{\rho}_i, \boldsymbol{\lambda}).$$

Note that the constraint of problem (2), namely, $\mathbf{x} \in \arg\min_{\hat{\mathbf{x}}} \sum_{i=0}^{\tau} g_i(\hat{\mathbf{x}}, \boldsymbol{\lambda})$, is equivalent to

$$\begin{aligned}\sum_{i=0}^{\tau} g_i(\mathbf{x}, \boldsymbol{\lambda}) &\le \min_{\mathbf{x}} \sum_{i=0}^{\tau} g_i(\mathbf{x}, \boldsymbol{\lambda}) \\ &= \max_{\boldsymbol{\rho}_i} -g_0^*\left(-\sum_{i=0}^{\tau}\boldsymbol{\rho}_i, \boldsymbol{\lambda}\right) - \sum_{i=1}^{\tau} g_i^*(\boldsymbol{\rho}_i, \boldsymbol{\lambda}).\end{aligned}$$

We can remove the max operator and find the identical constraint that

$$\sum_{i=0}^{\tau} g_i(\mathbf{x}, \boldsymbol{\lambda}) + g_0^*\left(-\sum_{i=1}^{\tau}\boldsymbol{\rho}_i, \boldsymbol{\lambda}\right) + \sum_{i=1}^{\tau} g_i^*(\boldsymbol{\rho}_i, \boldsymbol{\lambda}) \le 0.$$

The result is summarized in the following theorem.

Theorem 2.1. *Given convex, lower semi-continuous functions f and g_i, if $\bigcap_{i=0}^{\tau}\text{ri}(\text{dom } g_i) \neq \emptyset$, then Problem (2) has the following equivalent form:*

$$\begin{aligned}\min_{\mathbf{x},\boldsymbol{\rho}_i\in\mathbb{R}^n,\boldsymbol{\lambda}\in\mathcal{D}}\quad & f(\mathbf{x}, \boldsymbol{\lambda}) \\ \text{s.t.}\quad & \sum_{i=0}^{\tau} g_i(\mathbf{x}, \boldsymbol{\lambda}) + g_0^*\left(-\sum_{i=1}^{\tau}\boldsymbol{\rho}_i, \boldsymbol{\lambda}\right) \\ & + \sum_{i=1}^{\tau} g_i^*(\boldsymbol{\rho}_i, \boldsymbol{\lambda}) \le 0.\end{aligned} \tag{5}$$

The main benefit of reformulation (5) is circumventing the computation of complex value functions. Instead, it reduces to calculate the conjugate of each atom function g_i respectively, which has closed-form expression in many practical problems. We then demonstrate the power of this reformulation in hyperparameter selection problems. As a straightforward application of Theorem 2.1, (1) is equivalent to

$$\begin{aligned}\min_{\mathbf{x},\boldsymbol{\rho}_i\in\mathbb{R}^n,\boldsymbol{\lambda}\in\mathbb{R}_+^{\tau}}\quad & L(\mathbf{x}) \\ \text{s.t.}\quad & F(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\rho}) + \sum_{i=1}^{\tau}\lambda_i P_i(\mathbf{x}) \le 0,\end{aligned} \tag{6}$$

where we use the conventions $0P_i^*(\frac{\boldsymbol{\rho}_i}{0}) = 0$[2] and

$$F(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\rho}) = l(\mathbf{x}) + l^*\left(-\sum_{i=1}^{\tau}\boldsymbol{\rho}_i\right) + \sum_{i=1}^{\tau}\lambda_i P_i^*\left(\frac{\boldsymbol{\rho}_i}{\lambda_i}\right). \tag{7}$$

By introducing an auxiliary variables r_i satisfying $P_i(\mathbf{x}) \le r_i$, since $\lambda_i \ge 0$, constraint (6) is equivalent to

$$\begin{aligned}&F(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\rho}) + \sum_{i=1}^{\tau}\lambda_i r_i \le 0, \\ &P_i(\mathbf{x}) \le r_i \text{ for } i \in [\tau].\end{aligned}$$

This directly gives the following result.

Proposition 2.2. *Problem (1) can be reformulated as the following problem.*

$$\begin{aligned}\min_{\mathbf{x},\boldsymbol{\rho}_i\in\mathbb{R}^n,\mathbf{r},\boldsymbol{\lambda}\in\mathbb{R}_+^{\tau}}\quad & L(\mathbf{x}) \\ \text{s.t.}\quad & F(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\rho}) + \textstyle\sum_{i=1}^{\tau}\lambda_i r_i \le 0, \\ & P_i(\mathbf{x}) \le r_i,\ i \in [\tau], \quad \boldsymbol{\lambda} \ge 0.\end{aligned} \tag{8}$$

[1]Here, $\text{ri}(\cdot)$ denotes the relative interior of the set $\cdot$.

[2]By definition, $\lambda_i P_i^*(\frac{\boldsymbol{\rho}_i}{\lambda_i}) = \max_{\mathbf{z}_i} \boldsymbol{\rho}_i^T\mathbf{z}_i - \lambda_i P(\mathbf{z}_i)$. When $\lambda_i = 0$, $\lambda_i P_i^*(\frac{\boldsymbol{\rho}_i}{\lambda_i}) = \max_{\mathbf{z}_i} \boldsymbol{\rho}_i^T\mathbf{z}_i$ is 0 if $\boldsymbol{\rho}_i = 0$ and ∞ otherwise. The latter case contradicts strong duality and thus is abandoned.

Note that the function $F(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\rho})$ is convex because the conjugate functions l^* and P_i^* are convex, and $\lambda_i P_i^*\left(\frac{\rho_i}{\lambda_i}\right)$ is convex as it is the perspective of function $P_i^*(\boldsymbol{\rho}_i)$ (Boyd et al., 2004). We remark that the reformulation in Proposition 2.2 for (8) is proposed for the first time. The advantage of this reformulation is that it is a single level problem and does not involve any implicit function like the value function of the LL problem.

3 Majorization Minimization Algorithm for Hyperparameter Selection

In this section, we describe an algorithm that utilizes the reformulation (8).

3.1 Approximation via Majorization Function

Note that the only nonconvex term in (8) is the bilinear term $\sum_{i=1}^{\tau} \lambda_i r_i$. We adopt a majorization and minimization technique to handle this nonconvex term (Lange, 2016). To this end, we define a majorization function as follows.

Definition 3.1. We say $m(\bullet, \bullet; \bar{\xi}, \bar{\zeta})$ is a majorization for the bilinear form $\xi\zeta$ at $(\bar{\xi}, \bar{\zeta})$ if it satisfies

1. $m(\xi, \zeta; \bar{\xi}, \bar{\zeta}) \geq \xi\zeta$ and $m(\bar{\xi}, \bar{\zeta}; \bar{\xi}, \bar{\zeta}) = \bar{\xi}\bar{\zeta}$;
2. $m(\xi, \zeta; \bar{\xi}, \bar{\zeta})$ is a continuously differentiable function for (ξ, ζ), and $\frac{\partial m(\xi,\zeta;\bar{\xi},\bar{\zeta})}{\partial \xi} |_{(\xi,\zeta)=(\bar{\xi},\bar{\zeta})} = \bar{\zeta}$, $\frac{\partial m(\xi,\zeta;\bar{\xi},\bar{\zeta})}{\partial \zeta} |_{(\xi,\zeta)=(\bar{\xi},\bar{\zeta})} = \bar{\xi}$;
3. $\frac{\partial m(\xi,\zeta;\bar{\xi},\bar{\zeta})}{\partial \xi}$ and $\frac{\partial m(\xi,\zeta;\bar{\xi},\bar{\zeta})}{\partial \zeta}$ are locally Lipschitz continuous with respect to $(\bar{\xi}, \bar{\zeta})$.

There are various ways to construct such majorizations. For instance, when $\bar{\xi}, \bar{\zeta} > 0$, we can set

$$m(\xi, \zeta; \bar{\xi}, \bar{\zeta}) = \frac{1}{2}\left(\frac{\bar{\xi}}{\bar{\zeta}}\zeta^2 + \frac{\bar{\zeta}}{\bar{\xi}}\xi^2\right) \tag{9}$$

by using the Cauchy inequality. Another method is to use the identity

$$\xi\zeta = \frac{1}{4}(\xi + \zeta)^2 - \frac{1}{4}(\xi - \zeta)^2,$$

and set

$$m(\xi, \zeta; \bar{\xi}, \bar{\zeta}) = \frac{1}{4}(\xi+\zeta)^2 + \frac{1}{4}(\bar{\xi}-\bar{\zeta})^2 - \frac{1}{2}(\bar{\xi}-\bar{\zeta})(\xi-\zeta) \tag{10}$$

by linearizing the second term in the above identity at $(\bar{\xi}, \bar{\zeta})$.

Let m be a majorization of $\xi\zeta$ according to Definition 3.1. We now have the following inner approximation of (8) at $(\boldsymbol{\lambda}^k, \mathbf{r}^k)$,

$$\begin{array}{ll} \min\limits_{\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\rho},\mathbf{r}} & L(\mathbf{x}) \\ \text{s.t.} & F(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\rho}) + \sum_{i=1}^{\tau} m(\lambda_i, r_i; \lambda_i^k, r_i^k) \leq 0, \\ & P_i(\mathbf{x}) \leq r_i,\ i \in [\tau], \quad \boldsymbol{\lambda} \geq 0. \end{array} \tag{11}$$

Traditional MM algorithms solve the convex problem (11) iteratively. However, we point out that the above problem does not satisfy general constraint qualifications(CQs) like the Slater condition, which requires that there exists an interior point in the feasible region. Indeed, according to Proposition 2.2 and item 2 of Definition 3.1, we obtain that $F(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\rho}) + \sum_{i=1}^{\tau} m(\lambda_i, r_i; \lambda_i^k, r_i^k) \geq 0$ for any feasible solution, and thus there does not exist any interior point.

The absence of CQ not only prevents the use of general interior point methods for efficiently solving (8) (Wright et al., 1999), but also makes it difficult to show the convergence of solutions by sequentially solving (8) to KKT points (Andreani et al., 2016). To address this, we add a small positive number ϵ to the right-hand side of the first constraint in (8), and obtain the following approximation problem

$$\begin{array}{ll} \min\limits_{\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\rho},\mathbf{r}} & L(\mathbf{x}) + \frac{\beta}{2}\left\|\left(\mathbf{x}-\mathbf{x}^k, \boldsymbol{\lambda}-\boldsymbol{\lambda}^k, \mathbf{r}-\mathbf{r}^k, \boldsymbol{\rho}-\boldsymbol{\rho}^k\right)\right\|^2 \\ \text{s.t.} & F(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\rho}) + \sum_{i=1}^{\tau} m(\lambda_i, r_i; \lambda_i^k, r_i^k) \leq \epsilon, \\ & P_i(\mathbf{x}) \leq r_i,\ i \in [\tau], \quad \boldsymbol{\lambda} \geq 0. \end{array} \tag{12}$$

Here we also add a proximal term in the objective function to ensure the convergence of our algorithm. We summarise our method in Algorithm 1. We remark that line 1 of Algorithm 1 helps us find a feasible solution for problem (12), which guarantees the feasibility of problem (13), thanks to Definition 3.1.

Algorithm 1 Lower-level Dual based Majorization Minimization algorithm (LDMMA)

Require: initial $\epsilon > 0, \beta > 0$, and $\boldsymbol{\lambda}^0 \geq 0$.
1: Solve the lower-level subproblem $\min_{\mathbf{x}} l(\mathbf{x}) + \sum_{i=1}^{\tau} \lambda_i^0 P_i(\mathbf{x})$ and set $r_i^0 = P_i(\mathbf{x})$, $i = 1, 2, \ldots, \tau$
2: **for** $k = 0, 1, \ldots,$ **do**
3: Solve problem (12) and obtain an optimal solution $(\mathbf{x}^{k+1}, \mathbf{r}^{k+1}, \boldsymbol{\lambda}^{k+1}, \boldsymbol{\rho}^{k+1})$
4: **if** Termination criteria is met **then**
5: Stop
6: **end if**
7: **end for**

3.2 Conic Formulations of Subproblems

We point out that for all hyperparameter selection problems in Table 1, the subproblems of (11) or (12)

have explicit conic convex formulations, which can be solved by existing off-the-shelf solvers efficiently.

Here, we give an example of the elastic net problem. Other problems in Table 1 admit similar conic reformulations. We note that the full row rank condition is not necessary for the conic reformulation. Without such a condition, we can still obtain a conic program for the subproblem but with one extra linear constraint. See Appendix B for proofs, remarks, and more details on other problems.

Proposition 3.2. *Consider the elastic net problem with training data $A_{tr}, \mathbf{b}_{tr}$ and validation data $A_{val}, \mathbf{b}_{val}$,*

$$\begin{array}{ll}\min\limits_{\mathbf{x}} & L(\mathbf{x}) = \frac{1}{2}\|A_{val}\mathbf{x} - \mathbf{b}_{val}\|_2^2 \\ \text{s.t.} & \mathbf{x} \in \operatorname{argmin} \frac{1}{2}\|A_{tr}\mathbf{x} - \mathbf{b}_{tr}\|_2^2 + \lambda_1\|\mathbf{x}\|_1 + \frac{\lambda_2}{2}\|\mathbf{x}\|_2^2.\end{array}$$

If A_{tr} is of full row rank, then using (9) or (10), we obtain that the subproblem (11) for the above problem can be reformulated into the following conic program:

$$\begin{array}{ll}\min\limits_{\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\rho},\mathbf{r},t} & t \\ \text{s.t.} & A_{tr}^T\mathbf{w} + \boldsymbol{\rho}_1 + \boldsymbol{\rho}_2 = \mathbf{0}, \\ & \|\mathbf{x}\|_1 \le r_1,\ \|\boldsymbol{\rho}_1\|_\infty \le \lambda_1, \\ & SOCs(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\rho}, \mathbf{r}, t),\end{array}$$

where $SOCs(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\rho}, \mathbf{r}, t)$ represents second-order cone constraints with variables $(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\rho}, \mathbf{r}, t)$.

Equipped with the conic reformulations for the subproblems, one can take clear and concrete steps to apply Algorithm 1 to solve the hyperparameter selection problems. This enhances the implementability of the proposed algorithm, making it practical for real-world applications.

Before we end this section, we would like to emphasize the differences between our approach and the duality-based method in Ouattara and Aswani (2016). First, the main novelty of Ouattara and Aswani (2016) is to use the Lagrangian duality of the LL problem to deal with the constraints of LL problems, while we focus on Fenchel's duality for unconstrained LL problems. Second, the duality approach in Ouattara and Aswani (2016) still necessitate the calculation of an abstract value function h. In contrast, we utilize the splitting structures to obtain a reformulation that only consists of primal atom functions and their conjugates. Our approach circumvents the computation of complex value functions. Third, our reformulation leads to implementable subproblems in the form of conic programs for many problems of interest while that of Ouattara and Aswani (2016) does not.

4 Theoretical Investigations

In this section, we show that the sequence generated by Algorithm 1 converges to a KKT point of the ϵ-approximate problem

$$\begin{array}{ll}\min\limits_{\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\rho},\mathbf{r}} & L(\mathbf{x}) \\ \text{s.t.} & F(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\rho}) + \sum_{i=1}^{\tau} \lambda_i r_i \le \epsilon, \\ & P_i(\mathbf{x}) \le r_i,\ i \in [\tau], \quad \boldsymbol{\lambda} \ge 0.\end{array} \tag{13}$$

Note that in the above problem, we add a positive ϵ to the LL constraint. We remark that similar techniques are widely used in value function approaches in the literature Liu et al. (2021); Ye et al. (2022).

We begin with formal definitions of the KKT point and a nonsmooth CQ. Let $N_{\mathcal{X}}(\mathbf{x})$ denote the normal cone of the set $\mathcal{X}$ and $\partial\varphi$ denote the limiting sub-differential of the function φ (Rockafellar and Wets, 2009).

Definition 4.1. For a constrained optimization

$$\min_{\mathbf{x}\in\mathcal{X}} \hat{f}(\mathbf{x}) \qquad \text{s.t. } \hat{h}_i(\mathbf{x}) \le 0, \quad i = 1, 2, \ldots, m, \tag{14}$$

we say that $\mathbf{x}^*$ is its KKT point if there exists $\boldsymbol{\mu}^* \in \mathbb{R}^+$ such that $\mu_i^*\hat{h}_i(\mathbf{x}^*) = 0$, $\hat{h}_i(\mathbf{x}^*) \le 0$ and

$$\mathbf{0} \in \partial\hat{f}(\mathbf{x}^*) + \sum_{i=1}^{m} \mu_i^*\partial\hat{h}_i(\mathbf{x}^*) + N_{\mathcal{X}}(\mathbf{x}^*).$$

The following CQ is the nonsmooth version of the MFCQ that is frequently used for many algorithms.

Definition 4.2 (Jourani (1994); Ye et al. (2022))**.** Let $\mathbf{x}^*$ be a feasible point of (14). We say that the nonzero abnormal multiplier constraint qualification (NNAMCQ) holds at $\mathbf{x}^*$ for problem (14) if $\hat{h}_i(\mathbf{x}^*) < 0$ for $i \in [m]$ or $\mathbf{0} \notin$

$$\left\{\sum_{i=1}^{m} \mu_i\partial\hat{h}_i(\mathbf{x}^*) + N_{\mathcal{X}}(\mathbf{x}^*) : \mu_i\hat{h}_i(\mathbf{x}^*) = 0, \mu_i \ge 0, \boldsymbol{\mu} \ne \mathbf{0}\right\}.$$

Lemma 4.3 (NNAMCQ)**.** (i) *Let $\mathbf{x}^*$ be a solution of (14). If NNAMCQ holds at $\mathbf{x}^*$, then $\mathbf{x}^*$ is a KKT point of problem (14).* (ii) *NNAMCQ holds at any feasible point for problem (13).*

Let $\mathbf{z}^k := (\mathbf{x}^k, \boldsymbol{\lambda}^k, \mathbf{r}^k, \boldsymbol{\rho}^k)$ be the k-th iteration point of Algorithm 1. We use the following notations for the concerned problem (13) and its subproblem (12):

$$\begin{array}{rl}\mathcal{X} & = \{(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{r}, \boldsymbol{\rho}) : \boldsymbol{\rho} = (\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_\tau), \boldsymbol{\lambda}, \mathbf{r} \ge \mathbf{0}\}, \\ \mathbf{z} & = (\mathbf{x}, \boldsymbol{\lambda}, \mathbf{r}, \boldsymbol{\rho}) \in \mathcal{X}, \\ f(\mathbf{z}) & = L(\mathbf{x}), \\ f^k(\mathbf{z}) & = L(\mathbf{x}) + \frac{\beta}{2}(\|\mathbf{r} - \mathbf{r}^k\|^2 + \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^k\|^2), \\ g(\mathbf{z}) & = F(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\rho}) + \sum_{i=1}^{\tau} \lambda_i r_i - \epsilon, \\ \bar{g}^k(\mathbf{z}) & = F(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\rho}) + \sum_{i=1}^{\tau} m(\lambda_i, r_i; \lambda_i^k, r_i^k) - \epsilon, \\ h_j(\mathbf{z}) & = P_i(\mathbf{x}) - r_i,\ i = 1, 2, \ldots, \tau.\end{array}$$

By the definition of m in Definition 3.1, the following lemma naturally holds.

Lemma 4.4. *For $k = 0, 1, 2, \ldots$, we have the following results:* (i) $\bar{g}^k(\mathbf{z}) \geq g(\mathbf{z})$ *and* $\bar{g}^k(\mathbf{z}^k) = g(\mathbf{z}^k)$*;* (ii) $\partial \bar{g}^k(\mathbf{z}^k) = \partial g(\mathbf{z}^k)$.

We then introduce a sufficient decrease property of Algorithm 1.

Lemma 4.5. *Assume $L(\mathbf{x})$ is bounded below. Then for all $k \in \mathbb{N}$, we have* (i) $L(\mathbf{x}^{k+1}) - L(\mathbf{x}^k) \leq -\frac{\beta}{2}\|\mathbf{z}^{k+1} - \mathbf{z}^k\|^2$*;* (ii) $\lim_{k\to\infty} \|\mathbf{z}^{k+1} - \mathbf{z}^k\| = 0$.

Theorem 4.6. *Assume $L(\mathbf{x})$ is bounded below and $\{\mathbf{z}^k\}_{k\in\mathbb{N}}$ is bounded. The following two statements hold:*

(i) *If $\epsilon > 0$ in (13), then any accumulation point of $\{\mathbf{z}^k\}_{k\in\mathbb{N}}$ is a KKT point of (13);*

(ii) *Furthermore, if $L(\mathbf{x}), l(\mathbf{x})$, and $P_i(\mathbf{x})$, $i \in [\tau]$ are semi-algebraic functions, then $\{\mathbf{z}^k\}_{k\in\mathbb{N}}$ converges to a KKT point of (13).*

The ϵ perturbation in (13) is essential, which can be found in many value function based BLP algorithms (Ye et al., 2022; Gao et al., 2022; Xu and Ye, 2014). We suggest referring to Xu and Ye (2014); Ye et al. (2022) for the analysis of this relaxation. We remark that boundedness assumptions on $\{\mathbf{z}^k\}_{k\in\mathbb{N}}$ are widely used in relevant literature; see Ye et al. (2022) and Gao et al. (2022). We argue its necessity by referring to Theorem 4.2 in Attouch et al. (2013), a well-known convergence result requiring very mild conditions but needs the boundedness of the iterate sequence. We also remark that the convergence of our algorithm does not require the lower-level problem to be strongly convex, unlike many existing methods for BLP (Feng and Simon, 2018; Pedregosa, 2016).

5 Experiments

In this section, we conduct experiments to compare LDMMA with existing algorithms for hyperparameter optimization on synthetic data and real datasets, respectively. We briefly introduce our competitors in experiments:

- **Grid Search**: We perform a 10×10 uniformly-spaced grid search.
- **Random Search**: We uniformly sample 100 times for each direction of hyperparameters.
- **Implicit Differentiation**: We implement the IGJO algorithm in Feng and Simon (2018).
- **TPE**: We use the Tree-structured Parzen Estimator approach in Bergstra et al. (2013) which is known as a Bayesian optimization method.
- **VF-iDCA**: We implement the VF-iDCA algorithm in Gao et al. (2022), which considers the LL value function and applies DC program to approximately solve the BLP.

We consider hyperparameter optimization for elastic net, sparse group lasso, and support vector machines (Kunapuli et al., 2008; Feng and Simon, 2018; Gao et al., 2022). These three models only use a combination of regularization functions $\|\cdot\|_1, \|\cdot\|_2$ and $\frac{1}{2}\|\cdot\|_2^2$ that are included by our previous analysis. The elastic net (Zou and Hastie, 2003) is a linear combination of the lasso and ridge penalties and the sparse group lasso (Simon et al., 2013) combines the group lasso and lasso penalties, which are designed to encourage sparsity and grouping of predictors (Feng and Simon, 2018). The support vector machine is a classical machine learning model that assigns labels to objects (Noble, 2006) and its related BLP has been intensively studied (Kunapuli et al., 2008; Couellan and Wang, 2015; Jiang and Siddiqui, 2020). To compare the performance of each method, we calculate validation and test error with obtained LL minimizers from solving subproblems in each experiment. Our competitors are implemented using code from `https://github.com/SUSTech-Optimization?tab=repositories`. We use the off-the-shelf solver MOSEK[3] to solve the subproblem (12) at each iteration of LDMMA. The formulations of the three models and their associated subproblems can be found in Appendix B.

5.1 Experiments on Synthetic Data

The synthetic data consists of observation matrix samples from specific distribution and response vectors with rational noise. Detailed descriptions of the synthetic data generation settings and parameter settings of each method are in Appendix C.

5.1.1 Elastic Net

The numerical results on elastic net are reported in Table 2. We conduct 30 repeated experiments in each data size and take the average. Overall, LDMMA achieves the highest solution quality in the shortest running time on this problem model. Traditional gradient-free methods (grid search, random search, and TPE) still suffer from limitations in testing error and expensive time costs. Gradient-based methods IGJO perform slightly better on accuracy and efficiency, and VF-iDCA is the best among existing methods in the literature. Furthermore, LDMMA achieves more exquisite validation and test error beyond the reach of other methods along with greatly reduced time cost.

[3] `https://docs.mosek.com/9.3/toolbox/index.html`

Table 2: Elastic net problems on synthetic data, where $|I_{tr}|$, $|I_{val}|$, $|I_{te}|$ and p represent the number of training observations, validation observations, predictors and features, respectively.

Settings	Methods	Time(s)	Val. Err.	Test Err.	Settings	Time(s)	Val. Err.	Test Err.
	Grid	6.14 ± 2.34	6.31 ± 0.84	6.55 ± 0.91		7.49 ± 0.26	7.40 ± 1.29	6.05 ± 1.06
$\|I_{tr}\| = 100$	Random	9.54 ± 0.28	5.98 ± 2.24	6.56 ± 0.89	$\|I_{tr}\| = 100$	17.12 ± 0.40	7.40 ± 1.31	7.10 ± 1.08
$\|I_{val}\| = 20$	TPE	10.10 ± 0.44	6.05 ± 1.35	6.53 ± 0.90	$\|I_{val}\| = 100$	17.86 ± 0.92	7.38 ± 1.30	7.06 ± 1.06
$\|I_{te}\| = 250$	IGJO	3.92 ± 2.42	4.46 ± 1.75	6.76 ± 0.97	$\|I_{te}\| = 250$	4.02 ± 2.99	5.63 ± 1.24	5.36 ± 1.07
$p = 250$	VF-iDCA	0.84 ± 0.25	2.14 ± 0.76	4.03 ± 0.65	$p = 450$	2.57 ± 0.96	3.64 ± 0.53	4.73 ± 0.69
	LDMMA	0.64 ± 0.20	2.06 ± 0.42	3.91 ± 0.63		1.85 ± 0.21	3.15 ± 0.32	4.25 ± 0.48
	Grid	9.71 ± 0.21	6.82 ± 1.14	6.55 ± 0.91		13.17 ± 3.43	7.81 ± 1.53	8.82 ± 0.92
$\|I_{tr}\| = 100$	Random	9.54 ± 0.28	6.31 ± 0.84	6.68 ± 1.13	$\|I_{tr}\| = 100$	15.29 ± 2.60	6.44 ± 1.53	8.67 ± 0.94
$\|I_{val}\| = 100$	TPE	10.10 ± 0.44	6.30 ± 0.85	6.54 ± 1.15	$\|I_{val}\| = 100$	22.42 ± 1.30	7.71 ± 1.32	8.43 ± 0.80
$\|I_{te}\| = 250$	IGJO	3.92 ± 2.42	4.36 ± 0.96	5.54 ± 0.82	$\|I_{te}\| = 100$	31.30 ± 6.41	7.78 ± 1.12	8.61 ± 0.82
$p = 250$	VF-iDCA	1.90 ± 0.56	3.04 ± 1.51	4.52 ± 0.62	$p = 2500$	23.57 ± 4.06	1.83 ± 0.71	5.13 ± 1.02
	LDMMA	1.12 ± 0.15	2.63 ± 0.41	4.05 ± 0.97		9.30 ± 2.61	2.25 ± 1.09	4.19 ± 0.76

Table 3: Sparse group lasso problems on synthetic data, where p and M represent the number of covariates and covariate groups, respectively, and n represent the data scale described above.

Settings	Methods	Time(s)	Val. Err.	Test Err.	Settings	Time(s)	Val. Err.	Test Err.
	Grid	35.68 ± 1.85	43.80 ± 7.31	45.43 ± 7.87		45.72 ± 4.88	39.58 ± 5.31	46.66 ± 5.33
$n = 300$	Random	26.32 ± 1.51	36.94 ± 7.01	43.54 ± 8.87	$n = 450$	58.58 ± 1.24	43.91 ± 4.90	41.08 ± 9.05
$p = 600$	IGJO	49.00 ± 4.11	38.90 ± 6.21	41.94 ± 6.73	$p = 900$	64.90 ± 10.63	29.90 ± 7.15	48.82 ± 6.74
$M = 30$	VF-iDCA	8.69 ± 1.25	0.04 ± 0.01	37.31 ± 4.01	$M = 60$	25.41 ± 1.56	20.19 ± 6.04	36.36 ± 5.45
	LDMMA	6.85 ± 0.74	22.94 ± 2.56	21.25 ± 4.63		20.15 ± 2.61	21.04 ± 2.99	28.83 ± 6.76
	Grid	40.84 ± 1.04	42.45 ± 7.67	44.56 ± 7.33		74.22 ± 8.89	50.52 ± 4.14	59.90 ± 9.01
$n = 300$	Random	66.58 ± 1.01	39.27 ± 7.32	43.00 ± 8.83	$n = 600$	72.15 ± 4.49	53.21 ± 7.64	57.84 ± 14.52
$p = 900$	IGJO	60.67 ± 5.77	28.32 ± 4.93	43.43 ± 7.44	$p = 1200$	80.52 ± 5.66	41.70 ± 5.37	56.01 ± 12.74
$M = 60$	VF-iDCA	31.75 ± 5.62	17.85 ± 3.27	32.65 ± 4.83	$M = 150$	33.57 ± 7.48	25.64 ± 6.35	29.55 ± 3.88
	LDMMA	24.78 ± 0.92	24.54 ± 3.77	24.91 ± 3.58		27.34 ± 3.73	20.94 ± 3.52	23.74 ± 2.01

Table 4: Support Vector Machine problems with 3-fold and 6-fold cross-validation on three datasets, where the number of features p and samples $|\Omega|, |\Omega_{test}|$ are displayed together with dataset names. Results on other datasets are presented in Appendix C.

Dataset	Methods	3-fold			6-fold		
		Times(s)	Val. Err.	Test Err.	Times(s)	Val. Err.	Test Err.
diabetes-scale	Grid	3.17 ± 0.08	0.55 ± 0.03	0.19 ± 0.03	6.22 ± 0.21	0.54 ± 0.03	0.33 ± 0.04
$p = 8$	Random	3.47 ± 0.14	0.56 ± 0.03	0.32 ± 0.05	7.18 ± 0.30	0.55 ± 0.04	0.30 ± 0.05
$\|\Omega\| = 384$	TPE	10.21 ± 6.68	0.55 ± 0.04	0.29 ± 0.06	76.67 ± 36.39	0.54 ± 0.03	0.34 ± 0.06
$\|\Omega_{test}\| = 384$	VF-iDCA	0.28 ± 0.04	0.48 ± 0.03	0.23 ± 0.01	0.65 ± 0.03	0.43 ± 0.03	0.23 ± 0.02
	LDMMA	0.22 ± 0.03	0.49 ± 0.02	0.19 ± 0.01	0.55 ± 0.10	0.39 ± 0.05	0.20 ± 0.02
breast-cancer-scale	Grid	3.32 ± 0.09	0.08 ± 0.01	0.16 ± 0.08	6.32 ± 0.11	0.08 ± 0.01	0.15 ± 0.12
$p = 14$	Random	3.69 ± 0.07	0.09 ± 0.01	0.08 ± 0.08	7.20 ± 0.12	0.09 ± 0.02	0.10 ± 0.11
$\|\Omega\| = 336$	TPE	17.88 ± 10.05	0.09 ± 0.01	0.10 ± 0.11	34.66 ± 20.57	0.09 ± 0.01	0.18 ± 0.13
$\|\Omega_{test}\| = 347$	VF-iDCA	0.24 ± 0.04	0.09 ± 0.01	0.04 ± 0.01	0.57 ± 0.12	0.08 ± 0.01	0.03 ± 0.01
	LDMMA	0.12 ± 0.01	0.08 ± 0.01	0.03 ± 0.01	0.42 ± 0.17	0.08 ± 0.01	0.02 ± 0.01
w1a	Grid	20.08 ± 0.33	0.59 ± 0.10	0.41 ± 0.14	104.47 ± 2.99	0.06 ± 0.01	0.03 ± 0.00
$p = 300$	Random	20.30 ± 0.18	0.55 ± 0.07	0.31 ± 0.08	147.88 ± 8.64	0.05 ± 0.00	0.02 ± 0.00
$\|\Omega\| = 1236$	TPE	85.80 ± 13.95	0.64 ± 0.13	0.45 ± 0.11	682.35 ± 17.52	0.06 ± 0.01	0.03 ± 0.00
$\|\Omega_{test}\| = 1241$	VF-iDCA	4.32 ± 0.23	0.03 ± 0.02	0.03 ± 0.00	25.37 ± 3.10	0.01 ± 0.00	0.03 ± 0.00
	LDMMA	2.19 ± 0.24	0.01 ± 0.00	0.01 ± 0.00	15.25 ± 2.90	0.01 ± 0.00	0.02 ± 0.00

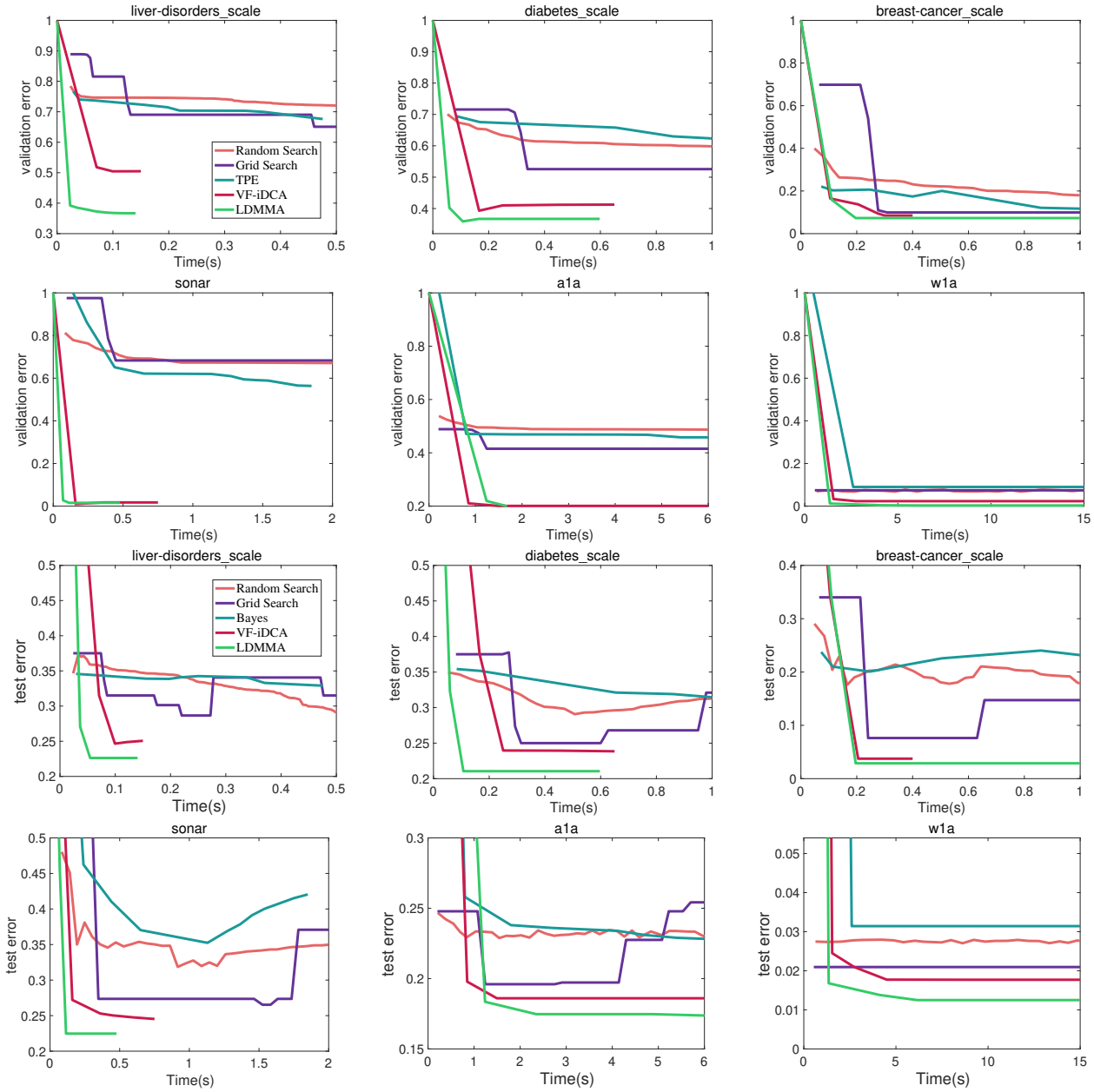

Figure 1: Comparison for variation trend of validation error and test error with time in SVM experiments.

Table 5: Elastic net problem on datasets gisette and sensit, where $|I_{tr}|$, $|I_{val}|$, $|I_{te}|$ and p represent the number of training samples, validation samples, test samples and features, respectively.

Settings	Methods	Time(s)	Val. Err.	Test Err.	Settings	Time(s)	Val. Err.	Test Err.
gisette	Grid	36.14 ± 4.52	0.22 ± 0.04	0.23 ± 0.01	sensit	1.39 ± 0.15	1.40 ± 0.79	1.31 ± 0.45
$p = 5000$	Random	55.44 ± 9.21	0.22 ± 0.05	0.23 ± 0.03	$p = 78823$	1.28 ± 0.10	1.52 ± 0.55	1.45 ± 0.40
$\|I_{tr}\| = 50$	TPE	40.01 ± 7.04	0.22 ± 0.05	0.24 ± 0.02	$\|I_{tr}\| = 25$	1.78 ± 0.09	1.38 ± 0.96	1.39 ± 0.55
$\|I_{val}\| = 50$	IGJO	6.15 ± 1.54	0.24 ± 0.05	0.24 ± 0.03	$\|I_{val}\| = 25$	0.49 ± 0.74	0.52 ± 0.20	0.61 ± 0.11
$\|I_{te}\| = 5900$	VF-iDCA	5.38 ± 1.65	0.00 ± 0.00	0.19 ± 0.01	$\|I_{te}\| = 50$	0.16 ± 0.08	0.23 ± 0.11	0.51 ± 0.06
	LDMMA	5.64 ± 0.94	0.00 ± 0.00	0.17 ± 0.01		0.16 ± 0.07	0.25 ± 0.12	0.40 ± 0.09

5.1.2 Sparse Group Lasso

We conduct experiments with different data scales and report numerical results averaged over 30 repetitions in Table 3. For each experiment, the generated datasets consist of n training, $n/3$ validation, and 100 test samples. LDMMA still performs the best in the sense that it achieves the minimum time cost and test error, meanwhile a similar validation error with VF-iDCA. As the dimension of data increases, our methods appear to offer all-round fitness for the problem, which indicates the superiority of LDMMA in large-scale hyperparameter optimization. It is worth noting that our algorithm can obtain the optimal solution for both hyperparameters and upper-level variables by solving problem (12). This is a significant advantage of our algorithm.

5.2 Experiments on Real Data

5.2.1 Support Vector Machine with Cross-validation

We conduct experiments for support vector machine (SVM) model on real-world datasets. Real-world datasets tend to be lager in size than synthetic datasets and exhibit more complex and irregular sample distributions. Consequently, hyperparameter selection will be heavily influenced by the partition of the training, validation, and test sets. Different partition can lead to substantial variations in the predictive performance of the models. Therefore, we perform 3-fold and 6-fold cross-validation using six moderately sized real datasets: liver-disorders, diabetes, breast-cancer, sonar, a1a (Asuncion and Newman, 2007), and w1a (Catanzaro et al., 2008). These datasets are derived from medical statistics and offer rich features and samples for analysis.

The details of corresponding subproblem for cross-validation with dataset partition and experimental settings are presented in Appendix C. We report numerical results on three datasets in Table 4 and Figure 1. As shown in Table 4, comparison results demonstrate that LDMMA consistently outperforms other optimization algorithms in terms of both the validation error and test error (except the case of diabetes-scal with 3-fold). Moreover, LDMMA achieves faster convergence than other methods. Figure 1 reports the variation trend of validation error and test error versus time from our experiments with 6-fold cross-validation. We emphasize that LDMMA remarkably reduces the validation and test errors at a faster speed than other algorithms. These results verify the superiority and applicability of our algorithm for SVM on real-world datasets.

5.2.2 Elastic Net with High Dimensional datasets

Furthermore, to certify the robustness of our algorithm, it is necessary to conduct experiments with larger scale which may capture more practical settings. We consider elastic net problem on high dimendional datasets gisette(Guyon et al., 2004) and sensit (Duarte and Hu, 2004). Experimental results are reported in Table 5, demonstrating that even in relatively high dimensional problems, LDMMA still achieves competitive performance at a fast speed.

6 Conclusion

In this paper, we propose a novel single-level reformulation for a group of hyperparameter optimization problems, where the main steps are leveraging the structure of the lower-level problem and applying Fenchel's duality. Our reformulation does not involve complex implicit functions but conjugates of some atom functions. Based on the new reformulation, we then propose the LDMMA, which applies the majorization-minimization method to obtain a convex subproblem. One superiority of our method is that for many practical problems, our subproblem are conic programs so that the subproblem can be efficiently solved by the off-the-shelf solvers. Theoretically, we prove the sequence convergence of the LDMMA. Numerical experiments on both synthetic and real-world data demonstrate the outperformance of LDMMA over existing methods.

We remark that the methods for solving subproblem (12) are not limited to the off-the-shelf solvers. In future work, we will explore first-order methods that are suitable for high-dimension settings for solving subproblem (12); see, e.g., Lan et al. (2011) and Necoara et al. (2019).

Acknowledgements

Rujun Jiang is partly supported by the National Key RD Program of China under grant 2023YFA1009300, National Natural Science Foundation of China under grants 12171100 and 72394364, and Natural Science Foundation of Shanghai 22ZR1405100. Anthony Man-Cho So is partly supported by the Hong Kong Research Grants Council (RGC) General Research Fund (GRF) project CUHK 14204823.

References

Andreani, R., Martinez, J. M., Ramos, A., and Silva, P. J. (2016). A cone-continuity constraint qualification and algorithmic consequences. *SIAM Journal on Optimization*, 26(1):96–110.

Asuncion, A. and Newman, D. (2007). Uci machine learning repository.

Attouch, H., Bolte, J., Redont, P., and Soubeyran, A. (2010). Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka–Łojasiewicz inequality. *Mathematics of Operations Research*, 35:438–457.

Attouch, H., Bolte, J., and Svaiter, B. F. (2013). Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming*, 137(1):91–129.

Bergstra, J., Yamins, D., and Cox, D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123. PMLR.

Bertrand, Q., Klopfenstein, Q., Blondel, M., Vaiter, S., Gramfort, A., and Salmon, J. (2020). Implicit differentiation of lasso-type models for hyperparameter optimization. In *International Conference on Machine Learning*, pages 810–821. PMLR.

Bertrand, Q., Klopfenstein, Q., Massias, M., Blondel, M., Vaiter, S., Gramfort, A., and Salmon, J. (2022). Implicit differentiation for fast hyperparameter selection in non-smooth convex learning. *The Journal of Machine Learning Research*, 23(1):6680–6722.

Bishop, C. M. et al. (1995). *Neural networks for pattern recognition*. Oxford university press.

Bolte, J., Daniilidis, A., and Lewis, A. (2007). The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223.

Bolte, J., Sabach, S., and Teboulle, M. (2014). Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494.

Boyd, S., Boyd, S. P., and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Brückner, M. and Scheffer, T. (2011). Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 547–555.

Catanzaro, B., Sundaram, N., and Keutzer, K. (2008). Fast support vector machine training and classification on graphics processors. In *Proceedings of the 25th international conference on Machine learning*, pages 104–111.

Chen, L., Xu, J., and Zhang, J. (2023). On bilevel optimization without lower-level strong convexity. *arXiv preprint arXiv:2301.00712*.

Claesen, M., De Smet, F., Suykens, J., and De Moor, B. (2014). Ensemblesvm: A library for ensemble learning using support vector machines. *arXiv preprint arXiv:1403.0745*.

Couellan, N. and Wang, W. (2015). Bi-level stochastic gradient for large scale support vector machine. *Neurocomputing*, 153:300–308.

Duarte, M. F. and Hu, Y. H. (2004). Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838.

Facchinei, F. and Pang, J.-S. (2003). *Finite-dimensional variational inequalities and complementarity problems*. Springer.

Feng, J. and Simon, N. (2018). Gradient-based regularization parameter selection for problems with nonsmooth penalty functions. *Journal of Computational and Graphical Statistics*, 27(2):426–435.

Franceschi, L., Donini, M., Frasconi, P., and Pontil, M. (2017). Forward and reverse gradient-based hyperparameter optimization. In *International Conference on Machine Learning*, pages 1165–1173. PMLR.

Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., and Pontil, M. (2018). Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR.

Gao, L. L., Ye, J., Yin, H., Zeng, S., and Zhang, J. (2022). Value function based difference-of-convex algorithm for bilevel hyperparameter selection problems. In *International Conference on Machine Learning*, pages 7164–7182. PMLR.

Guyon, I., Gunn, S., Ben-Hur, A., and Dror, G. (2004). Result analysis of the nips 2003 feature selection challenge. *Advances in neural information processing systems*, 17.

Jiang, W. and Siddiqui, S. (2020). Hyper-parameter optimization for support vector machines using stochastic gradient descent and dual coordinate descent. *EURO Journal on Computational Optimization*, 8(1):85–101.

Jourani, A. (1994). Constraint qualifications and lagrange multipliers in nondifferentiable programming problems. *Journal of Optimization Theory and Applications*, 81(3):533–548.

Kunapuli, G., Bennett, K. P., Hu, J., and Pang, J.-S. (2008). Classification model selection via bilevel programming. *Optimization Methods & Software*, 23(4):475–489.

Lan, G., Lu, Z., and Monteiro, R. D. (2011). Primal-dual first-order methods with iteration-complexity for cone programming. *Mathematical Programming*, 126(1):1–29.

Lange, K. (2016). *MM optimization algorithms*. SIAM.

Liu, B., Ye, M., Wright, S., Stone, P., et al. (2022). Bome! bilevel optimization made easy: A simple first-order approach. In *Advances in Neural Information Processing Systems*.

Liu, R., Liu, X., Yuan, X., Zeng, S., and Zhang, J. (2021). A value-function-based interior-point method for non-convex bi-level optimization. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6882–6892. PMLR.

Lorraine, J., Vicol, P., and Duvenaud, D. (2020). Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1552. PMLR.

Necoara, I., Patrascu, A., and Glineur, F. (2019). Complexity of first-order inexact lagrangian and penalty methods for conic convex programming. *Optimization Methods and Software*, 34(2):305–335.

Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567.

Ouattara, A. and Aswani, A. (2016). Duality approach to bilevel programs with a convex lower level. *2018 Annual American Control Conference (ACC)*, pages 1388–1395.

Pedregosa, F. (2016). Hyperparameter optimization with approximate gradient. In *International conference on machine learning*, pages 737–746. PMLR.

Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. (2019). Meta-learning with implicit gradients. *Advances in Neural Information Processing Systems*, 32.

Rockafellar, R. T. and Wets, R. J.-B. (2009). *Variational analysis*, volume 317. Springer Science & Business Media.

Shen, H. and Chen, T. (2023). On penalty-based bilevel gradient descent method. *arXiv preprint arXiv:2302.05185*.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *Journal of computational and graphical statistics*, 22(2):231–245.

Vial, J.-P. (1983). Strong and weak convexity of sets and functions. *Mathematics of Operations Research*, 8(2):231–259.

Wang, J., Chen, H., Jiang, R., Li, X., and Li, Z. (2021). Fast algorithms for stackelberg prediction game with least squares loss. In *Proceedings of the 38th International Conference on Machine Learning*, pages 10708–10716. PMLR.

Wang, J., Huang, W., Jiang, R., Li, X., and Wang, A. L. (2022). Solving stackelberg prediction game with least squares loss via spherically constrained least squares reformulation. In *ICML*. PMLR.

Wright, S., Nocedal, J., et al. (1999). Numerical optimization. *Springer Science*, 35(67-68):7.

Wu, Y. F., Zhang, W., Xu, P., and Gu, Q. (2020). A finite-time analysis of two time-scale actor-critic methods. *Advances in Neural Information Processing Systems*, 33:17617–17628.

Xu, M. and Ye, J. J. (2014). A smoothing augmented lagrangian method for solving simple bilevel programs. *Computational Optimization and Applications*, 59:353–377.

Yang, Z., Chen, Y., Hong, M., and Wang, Z. (2019). Provably global convergence of actor-critic: A case for linear quadratic regulator with ergodic cost. *Advances in Neural Information Processing Systems*, 32.

Ye, J. J., Yuan, X., Zeng, S., and Zhang, J. (2021). Difference of convex algorithms for bilevel programs with applications in hyperparameter selection. *arXiv preprint arXiv:2102.09006*.

Ye, J. J., Yuan, X., Zeng, S., and Zhang, J. (2022). Difference of convex algorithms for bilevel programs with applications in hyperparameter selection. *Mathematical Programming*, pages 1–34.

Zou, H. and Hastie, T. (2003). Regression shrinkage and selection via the elastic net, with applications to microarrays. *JR Stat Soc Ser B*, 67:301–20.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]

(b) Complete proofs of all theoretical results. [Yes]

(c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [Yes]

(b) The license information of the assets, if applicable. [Yes]

(c) New assets either in the supplemental material or as a URL, if applicable. [Yes]

(d) Information about consent from data providers/curators. [Yes]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

The appendix is organized as follows. In Appendix A, we provide missing proofs of Sec. 4. In Appendix. B, we use several widely used machine learning models to illustrate that the constraints in (11) can often be represented by conic inequalities. In Appendix C, we provide details of our numerical experiments and give some additional experiments results.

A Proofs in Section 4

A.1 Proof for Lemma 4.3

Proof. (i) The first item follows from Ye et al. (2022).

(ii) The second item can be proved by contradiction. If NNAMCQ fails at some feasible point $\mathbf{z} := (\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{r}, \rho)$, i.e., there exists vector $(\mu_0, \mu_1, \ldots, \mu_\tau)$ such that $0 \in \mu_0 \partial g(\mathbf{z}) + \sum_{i=1}^{\tau} \mu_i \partial h_j(\mathbf{z})$, one can obtain a contradiction whether the constraint $g(\mathbf{z}) = F(\mathbf{x}, \boldsymbol{\lambda}, \rho) + \sum_i \lambda_i r_i - \epsilon \leq 0$ is active or not.

1. When $F(\mathbf{x}, \boldsymbol{\lambda}, \rho) + \sum_i \lambda_i r_i - \epsilon < 0$, $\mu_0 = 0$ and we observe that $\partial h_i(\mathbf{z}) = \partial \left(P_i(\mathbf{x}) - r_i\right)$, $i \in [\tau]$ are linearly independent due to term r_i. Then $0 \in \mu_0 \partial g(\mathbf{z}) + \sum_{i=1}^{\tau} \mu_i \partial h_j(\mathbf{z})$ implies $\mu_i = 0$ for $i \in [\tau]$, which contradicts the definition of NNAMCQ that requires $\boldsymbol{\mu} \neq 0$.

2. When $F(\mathbf{x}, \boldsymbol{\lambda}, \rho) + \sum_i \lambda_i r_i - \epsilon = 0$ is active, the condition $0 \in \mu_0 \partial g(\mathbf{z}) + \sum_{i=1}^{\tau} \mu_i \partial h_j(\mathbf{z})$ implies $\mathbf{z} = \operatorname{argmin}_z F(\mathbf{x}, \boldsymbol{\lambda}, \rho) + \sum_i \lambda_i r_i$. This implies $F(\mathbf{x}, \boldsymbol{\lambda}, \rho) + \sum_i \lambda_i r_i = 0$ because the left hand side of the constraint in (5) is always larger than or equal to 0 due to strong duality of the lower-level problem. This contradicts $F(\mathbf{x}, \boldsymbol{\lambda}, \rho) + \sum_i \lambda_i r_i - \epsilon = 0$.

□

A.2 Proof for Lemma 4.5

Proof. We first prove that for $k \geq 0$, $\mathbf{z}^k$ is a feasible point of (12) for all $k \in \mathbb{N}$ by induction. Suppose this statement holds for some $k = d \geq 0$, we prove it holds for $k = d+1$. Note that $(\mathbf{x}^{d+1}, \boldsymbol{\lambda}^{d+1}, \mathbf{r}^{d+1}, \boldsymbol{\rho}^{d+1})$ is optimal for problem (12) with $\mathbf{r}^d$ and $\boldsymbol{\lambda}^d$. From the optimality of $\mathbf{z}^{d+1}$ to (12), we have

$$h_j(\mathbf{z}^{d+1}) \leq 0, \ \forall\ j \in J \quad \text{and} \quad \bar{g}^d(\mathbf{z}^{d+1}) \leq 0.$$

Since $\bar{g}^d(\mathbf{z}^{d+1}) \geq g(\mathbf{z}^{d+1}) = \bar{g}^{d+1}(\mathbf{z}^{d+1})$ due to Lemma 4.4, it follows that $\bar{g}^{d+1}(\mathbf{z}^{d+1}) \leq 0$ and thus $\mathbf{z}^{d+1}$ is feasible for problem (12) with $k = d+1$. Finally, $\mathbf{z}^0$ is obviously feasible for (12) with $k = 0$ due to our special choice of $\boldsymbol{\lambda}^0$ and $\mathbf{r}^0$.

Note that $(\mathbf{x}^{k+1}, \boldsymbol{\lambda}^{k+1}, \mathbf{r}^{k+1}, \boldsymbol{\rho}^{k+1})$ is optimal for problem (12). By the optimality of $\mathbf{z}^{k+1}$ and the feasibility of $\mathbf{z}^k$, we have

$$L(\mathbf{x}^{k+1}) + \frac{\beta}{2}\|\mathbf{z}^{k+1} - \mathbf{z}^k\|^2 \leq L(\mathbf{x}^k),$$

which proves (i). (Here $\mathbf{x}^0$ can be any feasible point.) Item (ii) directly follows from (i) and that L is bounded below. □

A.3 Proof for Theorem 4.6

Proof. (i) Assume subsequence $\{\mathbf{z}^{k_j}\}_{j \in \mathbb{N}}$ converges to $\mathbf{z}^*$. By Lemma 4.3, the NNAMCQ holds at $\mathbf{z}^{k_j}$ for problem (12) with $k = k_j - 1$. Moreover, the KKT optimality conditions implies that there exist acceptable Lagrange multipliers $\boldsymbol{\mu}^{k_j} = (\mu_0^{k_j}, \mu_1^{k_j} \ldots, \mu_\tau^{k_j})$ such that

$$\mu_0^{k_j} \bar{g}^{k_j}(\mathbf{z}^{k_j}) = 0, \quad \mu_i^{k_j} h_i(\mathbf{z}^{k_j}) = 0, \ i = 1, \ldots \tau$$

and

$$\mathbf{0} \in \partial f^{k_j - 1}(\mathbf{z}^{k_j}) + \mu_0^{k_j} \partial \bar{g}^{k_j - 1}(\mathbf{z}^{k_j}) + \sum_{i=1}^{\tau} \mu_i^{k_j} \partial h_i(\mathbf{z}^{k_j}) + N_{\mathcal{X}}(\mathbf{z}^{k_j}). \tag{15}$$

We then claim $\{\boldsymbol{\mu}^{l_j}\}_{j\in\mathbb{N}}$ is bounded. Since $\|\mathbf{r}^{l_j}-\mathbf{r}^{l_j-1}\|\to 0$ and $\|\boldsymbol{\lambda}^{l_j}-\boldsymbol{\lambda}^{l_j-1}\|\to 0$ by Lemma 4.5, from $\mathbf{z}^{l_j}\to\mathbf{z}^*$ and Definition 3.1 we have

$$\partial\sum_{i=1}^{\tau} m(\lambda_i,r_i;\lambda_i^{l_j-1},r_i^{l_j-1})\mid_{(\lambda_i^{l_j},r_i^{l_j})}\to \partial\sum_{i=1}^{\tau}\lambda_i r_i\mid_{(\lambda_i^*,r_i^*)}. \tag{16}$$

According to Exercise 8.8(c) in Rockafellar and Wets (2009) and the expressions of $\bar{g}^k$ and g, (16) yields

$$\limsup_{j\to\infty}\partial\bar{g}^{l_j-1}(\mathbf{z}^{l_j})\subseteq\partial g(\mathbf{z}^*).$$

If $\{\boldsymbol{\mu}^{l_j}\}_{j\in N}$ is unbounded, by passing to a further subsequence if necessary, we assume$\boldsymbol{\mu}^{l_j}/\|\boldsymbol{\mu}^{l_j}\|\to\bar{\boldsymbol{\mu}}$. Note that $\|\boldsymbol{\mu}^{l_j}\|\to\infty$ and $\partial f(\mathbf{z}^{l_j})$ is bounded. Multiplying $\frac{1}{\|\boldsymbol{\mu}^{l_j}\|}$ on the right-hand side of (15) and taking limit to (15), we have

$$\begin{aligned}&\mathbf{0}\in\bar{\mu}_0\partial g(\mathbf{z}^*)+\textstyle\sum_{i=1}^m\bar{\mu}_i\partial h_i(\mathbf{z}^*)+N_{\mathcal{X}}(\mathbf{z}^*)\text{ with }\|\bar{\boldsymbol{\mu}}\|=1,\\&\bar{\mu}_0 g(\mathbf{x}^*)=0,\ \bar{\mu}_i h_i(\mathbf{x}^*)=0, i\in[\tau],\ \bar{\mu}_i\geq 0,\ i=0,\ldots,\tau.\end{aligned}$$

which contradicts the NNAMCQ at $\mathbf{z}^*$. Therefore, $\{\boldsymbol{\mu}^{l_j}\}_{j\in N}$ is bounded.

Then we assume $\boldsymbol{\mu}^{k_j}\to\boldsymbol{\mu}^*$ by passing to a subsequence if necessary. Note that $\limsup_{j\to\infty}\partial f^{k_j}(\mathbf{z}^{k_j})\subseteq\partial f(\mathbf{z}^*)$ also holds because of the expression of f^k and the outer semi-continuity of $\partial L(\mathbf{x})$ (see, e.g., Definition 5.4 and Proposition 8.7 in Rockafellar and Wets (2009)). By $\mathbf{z}^{k_j}\to\mathbf{z}^*$, $\limsup_{j\to\infty}\partial\bar{g}^{k_j-1}(\mathbf{z}^{k_j})\subseteq\partial g(\mathbf{z}^*)$ and $\limsup_{j\to\infty}\partial f^{k_j}(\mathbf{z}^{k_j})\subseteq\partial f(\mathbf{z}^*)$, (15) yields

$$\mathbf{0}\in\partial f(\mathbf{z}^*)+\mu_0^*\partial g(\mathbf{z}^*)+\sum_{i=1}^{\tau}\mu_i^*\partial h_i(\mathbf{z}^*)+N_{\mathcal{X}}(\mathbf{z}^*),$$

which says that $\mathbf{z}^*$ is a KKT point of (13).

(ii) We construct a simple merit function for our model, which is defined by

$$G(\mathbf{z})=L(\mathbf{x})+\delta_{\mathcal{X}}(\mathbf{z})+\delta_{\{\mathbf{z}:g(\mathbf{z})\leq 0\}}(\mathbf{z})+\sum_{j=1}^{\tau}\delta_{\{\mathbf{z}:h_j(\mathbf{z})\leq 0\}}(\mathbf{z}).$$

By the feasibility of $\mathbf{z}^{k+1}$ in (12) and Definition 3.1, $h_j(\mathbf{z}^{k+1})\leq 0$, $j\in[\tau]$ and $g(\mathbf{z}^{k+1})\leq\bar{g}^k(\mathbf{z}^{k+1})\leq 0$. Hence,

$$G\left(\mathbf{z}^{k+1}\right)=L(\mathbf{x}^{k+1})\text{ for all }k\in\mathbb{N}.$$

This, together with $L(\mathbf{x}^{k+1})\leq L(\mathbf{x}^k)-\frac{\beta}{2}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2$ given by Lemma 4.5, yields

$$G\left(\mathbf{z}^{k+1}\right)\leq G\left(\mathbf{z}^k\right)-\frac{\beta}{2}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2. \tag{17}$$

We then establish the relative error, which needs to estimate $\operatorname{dist}(\mathbf{0},\partial G(\mathbf{z}))$. To begin with, by the convexity of $L(\mathbf{x})$ and $P_i(\mathbf{x})$, $L(\mathbf{x})$ and $h_j(\mathbf{z})$, $j\in[\tau]$ are regular. By the convexity of F and weak convexity of $\sum_{i=1}^{\tau}\lambda_i r_i$, $g(\mathbf{z})$ is regular due to Proposition 4.5 in Vial (1983). Then, $\delta_{\{\mathbf{z}:g(\mathbf{z})\leq 0\}}(\mathbf{z})$ and $\delta_{\{\mathbf{z}:h_j(\mathbf{z})\leq 0\}}(\mathbf{z})$, $j\in[\tau]$ are regular by Exercise 8.14 in Rockafellar and Wets (2009). It follows from Corollary 10.9 in Rockafellar and Wets (2009) that

$$\partial G(\mathbf{z})=\partial L(\mathbf{x})+\partial\delta_{\mathcal{X}}(\mathbf{z})+\partial\delta_{\{\mathbf{z}:g(\mathbf{z})\leq 0\}}(\mathbf{z})+\sum_{j=1}^{\tau}\partial\delta_{\{\mathbf{z}:h_j(\mathbf{z})\leq 0\}}(\mathbf{z}).$$

Since Exercise 8.14 in Rockafellar and Wets (2009) ensures that $\partial\delta_{\mathcal{X}}(\mathbf{z})=N_{\mathcal{X}}(\mathbf{z})$, $\partial\delta_{\{\mathbf{z}:g(\mathbf{z})\leq 0\}}(\mathbf{z})=N_{\{\mathbf{z}:g(\mathbf{z})\leq 0\}}(\mathbf{z})$, and $\partial\delta_{\{\mathbf{z}:h_j(\mathbf{z})\leq 0\}}(\mathbf{z})=N_{\{\mathbf{z}:h_j(\mathbf{z})\leq 0\}}(\mathbf{z})$, $j\in[\tau]$, we have

$$\partial G(\mathbf{z})=\partial L(\mathbf{x})+N_{\mathcal{X}}(\mathbf{z})+N_{\{\mathbf{z}:g(\mathbf{z})\leq 0\}}(\mathbf{z})+\sum_{j=1}^{\tau}N_{\{\mathbf{z}:h_j(\mathbf{z})\leq 0\}}(\mathbf{z}).$$

By Corollary 10.50 in Rockafellar and Wets (2009), it follows that $\partial G(\mathbf{z}) = \nabla L(\mathbf{x}) + N_{\mathcal{X}}(\mathbf{z}) + \mu_0 \partial g(\mathbf{z})(\mathbf{z}) + \sum_{j=1}^{\tau} \mu_i \partial h_j(\mathbf{z})$, where $\mu_0, \mu_j \geq 0$, $\mu_0 g(\mathbf{z}) = 0$, and $\mu_j h_j(\mathbf{z}) = 0$ for $j \in [\tau]$. Thus,

$$\begin{aligned} \partial G(\mathbf{z}^k) = \quad & \{\partial L(\mathbf{x}^k) + N_{\mathcal{X}}(\mathbf{z}^k) + \mu_0^k \partial g(\mathbf{z}^k) + \textstyle\sum_{j=1}^{\tau} \mu_i^k \partial h_j(\mathbf{z}^k) : \\ & \mu_0^k, \mu_j^k \geq 0, \mu_0^k g(\mathbf{z}^k) = 0, \mu_j^k h_j(\mathbf{z}^k) = 0 \text{ for } j \in [\tau]\}. \end{aligned} \tag{18}$$

Note that $\partial f^{k-1}(\mathbf{z}^k) = \partial L(\mathbf{x}^k) + \beta(\mathbf{z}^k - \mathbf{z}^{k-1})$ and that Definition 3.1 implies

$$\partial \bar{g}^{k-1}(\mathbf{z}^k) = \partial g(\mathbf{z}^k) + \sum_{i=1}^{\tau} \left(\nabla m(\lambda_i^k, r_i^k; \lambda_i^{k-1}, r_i^{k-1}) - \nabla m(\lambda_i^k, r_i^k; \lambda_i^k, r_i^k) \right).$$

Comparing (18) and the optimality conditions (15), we have

$$\mathbf{0} \in \partial G(\mathbf{z}^k) + \beta(\mathbf{z}^k - \mathbf{z}^{k-1}) + \mu_0^k \sum_{i=1}^{\tau} (\nabla m(\lambda_i^k, r_i^k; \lambda_i^{k-1}, r_i^{k-1}) - \nabla m(\lambda_i^k, r_i^k; \lambda_i^k, r_i^k)),$$

where μ_0^k is an acceptable Lagrange multiplier. This is equivalent to

$$\beta(\mathbf{z}^{k-1} - \mathbf{z}^k) + \sum_{i=1}^{\tau} \left(\nabla m(\lambda_i^k, r_i^k; \lambda_i^k, r_i^k) - \nabla m(\lambda_i^k, r_i^k; \lambda_i^{k-1}, r_i^{k-1}) \right) \in \partial G(\mathbf{z}^k).$$

By the triangle inequality, it follows that

$$\begin{aligned} & \operatorname{dist}\left(\mathbf{0}, \partial G(\mathbf{z}^k)\right) \\ \leq & \beta \left\| \mathbf{z}^{k-1} - \mathbf{z}^k \right\| + \mu_0^k \sum_{i=1}^{\tau} \left\| \nabla m(\lambda_i^k, r_i^k; \lambda_i^k, r_i^k) - \nabla m(\lambda_i^k, r_i^k; \lambda_i^{k-1}, r_i^{k-1}) \right\|. \end{aligned} \tag{19}$$

Recall that we have proved that sub-sequence $\{\mu^{k_j}\}_{j\in\mathbb{N}}$ is bounded in (i) by the NNAMFCQ if $\{\mathbf{z}^{k_j}\}_{j\in\mathbb{N}}$ converges. This, together with the boundedness of $\{\mathbf{z}^k\}_{k\in\mathbb{N}}$ ensures that $\{\mu^k\}_{k\in\mathbb{N}}$ is bounded. Thus,

$$\mu_0^k \leq M_1 \text{ for some } M_1 > 0. \tag{20}$$

Furthermore, by the local Lipschitz continuity of $\nabla m(\xi, \zeta; \bar{\xi}, \bar{\zeta})$ w.r.t. $(\bar{\xi}, \bar{\zeta})$ given by Definition 3.1, for $\{\mathbf{z}^k\}_{k\in\mathbb{N}}$, there exists $M_2 > 0$ such that

$$\begin{aligned} \|\nabla m(\lambda_i^k, r_i^k; \lambda_i^k, r_i^k) - \nabla m(\lambda_i^k, r_i^k; \lambda_i^{k-1}, r_i^{k-1})\| &\leq M_2 \|(\lambda_i^k - \lambda_i^{k-1}, r_i^k - r_i^{k-1})\| \\ &\leq M_2 \left\| \mathbf{z}^{k-1} - \mathbf{z}^k \right\|, \end{aligned} \tag{21}$$

where the second inequality directly follows from $\mathbf{z} = (\mathbf{x}, \boldsymbol{\lambda}, \mathbf{r}, \boldsymbol{\rho})$. Combining (19), (20), and (21), we have

$$\operatorname{dist}\left(\mathbf{0}, \partial G(\mathbf{z}^k)\right) \leq (\beta + \tau M_1 M_2) \left\| \mathbf{z}^{k-1} - \mathbf{z}^k \right\|. \tag{22}$$

Next, we verify that G satisfies the Kurdyka-Lojasiewicz (KL) property. It suffices to show that G is a semi-algebraic function by Bolte et al. (2007). Note that it has been shown by Attouch et al. (2010, 2013); Facchinei and Pang (2003); Bolte et al. (2014) that the semi-algebraic property is preserved under many operations such as finite sum, product, and partial maximization operations. Moreover, the epi-graphs of semi-algebraic functions are semi-algebraic sets and indicator functions of semi-algebraic sets are semi-algebraic functions. As a direct consequence, $L(\mathbf{x}), g(\mathbf{z})$, and $h_j(\mathbf{z}), j \in [\tau]$ are all semi-algebraic functions. $\mathcal{X}, \{\mathbf{z} : g(\mathbf{z}) \leq 0\}$, and $\{\mathbf{z} : h_j(\mathbf{z}) \leq 0\}$, $j \in [\tau]$ are semi-algebraic sets and $\delta_{\{\mathbf{z}:g(\mathbf{z})\leq 0\}}, \delta_{\{\mathbf{z}:h_j(\mathbf{z})\leq 0\}}, j \in [\tau]$ are semi-algebraic functions. Hence, G is semi-algebraic and satisfies the KL property.

Finally, combining (17), (22), and that G satisfies the KL property, Theorem 2.9 in Attouch et al. (2013) implies the convergence of $\{\mathbf{z}^k\}_{k\in\mathbb{N}}$ and that the limiting point $\mathbf{z}^*$ is a stationary of G, i.e., $\mathbf{0} \in \partial G(\mathbf{z}^*)$. By (18), this is equivalent to that $\mathbf{z}^*$ is a KKT point of (13). The proof is complete.

□

B Reformulations and Subproblems of LDMMA for Different Models

In this section, we consider several widely used machine learning models to illustrate that the constraints in (11) can often be represented by conic inequalities. We choose $m(\xi,\zeta;\bar{\xi},\bar{\zeta}) = \frac{1}{2}\left(\frac{\bar{\xi}}{\bar{\zeta}}\zeta^2 + \frac{\bar{\zeta}}{\bar{\xi}}\xi^2\right)$ to demonstrate the tractability of our subproblems when $\mathbf{r}^k, \boldsymbol{\lambda}^k > 0$[4].

B.1 Elastic Net and Sparse Group Lasso

We recall the elastic net problem

$$\begin{array}{ll}\min\limits_{\mathbf{x}} & L(\mathbf{x}) = \frac{1}{2}\|A_{val}\mathbf{x} - \mathbf{b}_{val}\|^2 \\ \text{s.t.} & \mathbf{x} \in \operatorname{argmin} \frac{1}{2}\|A_{tr}\mathbf{x} - \mathbf{b}_{tr}\|^2 + \lambda_1\|\mathbf{x}\|_1 + \frac{\lambda_2}{2}\|\mathbf{x}\|_2^2\end{array}$$

and the sparse group lasso problem

$$\begin{array}{ll}\min\limits_{\mathbf{x}} & L(\mathbf{x}) = \frac{1}{2}\|A_{val}\mathbf{x} - \mathbf{b}_{val}\|^2 \\ \text{s.t.} & \mathbf{x} \in \operatorname{argmin} \frac{1}{2}\|A_{tr}\mathbf{x} - \mathbf{b}_{tr}\|^2 + \lambda_{M+1}\|\mathbf{x}\|_1 + \sum\limits_{i=1}^{M}\lambda_i\|\mathbf{x}^{(i)}\|_2,\end{array}$$

where $\mathbf{x}^{(i)}$ is a sub-vector of $\mathbf{x}$.

The following two propositions show the explicit formulation of (11), which are closely related to (12), subproblems of the proposed LDMMA.

Proposition B.1. *For the elastic net problem with linearly independent $\{\mathbf{a}_i : i \in I_{tr}\}$, let $A_{tr} = (\mathbf{a}_i)_{i\in I_{tr}}, \mathbf{b}_{tr} = (b_i)_{i\in I_{tr}}$ denote the training data and $A_{val} = (\mathbf{a}_i)_{i\in I_{val}}, \mathbf{b}_{val} = (b_i)_{i\in I_{val}}$ denote the validation data, then (11) can be reformulated into the following conic program:*

$$\begin{array}{ll}\min & t \\ \text{s.t.} & \left\|\left(A_{tr}\mathbf{x} - \mathbf{b}_{tr}, \sqrt{\frac{\lambda_1^0}{r_1^0}}r_1, \sqrt{\frac{r_1^0}{\lambda_1^0}}\lambda_1, \sqrt{\frac{\lambda_2^0}{r_2^0}}r_2, \sqrt{\frac{r_2^0}{\lambda_2^0}}\lambda_2, \mathbf{w} + \mathbf{b}_{tr}, \frac{s}{\|\mathbf{b}_{tr}\|_2}\right)\right\|_2 \le \|\mathbf{b}_{tr}\|_2 - \frac{1}{\|\mathbf{b}\|_2}s, \\ & A_{tr}^T\mathbf{w} + \boldsymbol{\rho}_1 + \boldsymbol{\rho}_2 = \mathbf{0}, \\ & \|\mathbf{x}\|_1 \le r_1,\ \|\boldsymbol{\rho}_1\|_\infty \le \lambda_1, \\ & \left\|\begin{pmatrix}\mathbf{x} \\ r_2 - \frac{1}{2}\end{pmatrix}\right\|_2 \le r_2 + \frac{1}{2},\ \left\|\begin{pmatrix}\sqrt{2}\boldsymbol{\rho}_2 \\ s - \lambda_2\end{pmatrix}\right\|_2 \le s + \lambda_2, \\ & \left\|\begin{pmatrix}A_{val}\mathbf{x} - \mathbf{b}_{val} \\ t - \frac{1}{2}\end{pmatrix}\right\|_2 \le t + \frac{1}{2}.\end{array}$$

Proposition B.2. *For the sparse group lasso problem with linearly independent $\{\mathbf{a}_i : i \in I_{tr}\}$, let $A_{tr} = (\mathbf{a}_i)_{i\in I_{tr}}, \mathbf{b}_{tr} = (b_i)_{i\in I_{tr}}$ denote the training data and $A_{val} = (\mathbf{a}_i)_{i\in I_{val}}, \mathbf{b}_{val} = (b_i)_{i\in I_{val}}$ denote the validation data, then (11) can be reformulated into the following conic program:*

$$\begin{array}{ll}\min & t \\ \text{s.t.} & \left\|\left(A_{tr}\mathbf{x} - \mathbf{b}_{tr}, \sqrt{\frac{\lambda_1^0}{r_1^0}}r_1, \sqrt{\frac{r_1^0}{\lambda_1^0}}\lambda_1, \ldots \sqrt{\frac{\lambda_{M+1}^0}{r_{M+1}^0}}r_{M+1}, \sqrt{\frac{r_{M+1}^0}{\lambda_{M+1}^0}}\lambda_{M+1}, \mathbf{w} + \mathbf{b}_{tr}\right)\right\|_2 \le \|\mathbf{b}_{tr}\|_2, \\ & A_{tr}^T\mathbf{w} + \sum\limits_{i=1}^{M+1}\boldsymbol{\rho}_i = \mathbf{0}, \\ & \|\mathbf{x}\|_1 \le r_{M+1},\ \|\boldsymbol{\rho}_1\|_\infty \le \lambda_{M+1}, \\ & \|\mathbf{x}^{(i)}\|_2 \le r_i,\ \|\boldsymbol{\rho}_i\|_2 \le \lambda_i \quad \text{for } i = 1, 2, \ldots, M, \\ & \left\|\begin{pmatrix}A_{val}\mathbf{x} - \mathbf{b}_{val} \\ t - \frac{1}{2}\end{pmatrix}\right\|_2 \le t + \frac{1}{2}.\end{array}$$

As a comparison, the reformulation (8) for elastic net and sparse group lasso are respectively

$$\begin{array}{ll}\min & \frac{1}{2}\|A_{val}\mathbf{x} - \mathbf{b}_{val}\|_2^2 \\ \text{s.t.} & \frac{1}{2}\|A_{tr}\mathbf{x} - \mathbf{b}_{tr}\|_2^2 + \frac{1}{2}\|\mathbf{w} + \mathbf{b}\|_2^2 - \frac{1}{2}\|\mathbf{b}\|_2^2 + \lambda_1 r_1 + \lambda_2 r_2 + \frac{1}{2\lambda_2}\|\boldsymbol{\rho}_2\|_2^2 \le 0, \\ & A^T\mathbf{w} + \boldsymbol{\rho}_1 + \boldsymbol{\rho}_2 = \mathbf{0}, \\ & \|\mathbf{x}\|_1 \le r_1,\ \frac{1}{2}\|\mathbf{x}\|_2^2 \le r_2, \|\boldsymbol{\rho}\|_\infty \le \lambda_1, \quad \boldsymbol{\lambda} \ge 0\end{array} \tag{23}$$

[4]We note that for the case that $r_i^k = 0$ or $\lambda_i^k = 0$, we can choose m by (10) and conclude similar results.

and

$$\begin{array}{ll} \min & \frac{1}{2}\|A_{val}\mathbf{x}-\mathbf{b}_{val}\|_2^2 \\ \text{s.t.} & \frac{1}{2}\|A_{tr}\mathbf{x}-\mathbf{b}_{tr}\|_2^2+\frac{1}{2}\|\mathbf{w}+\mathbf{b}\|_2^2-\frac{1}{2}\|\mathbf{b}\|_2^2+\sum_{i=1}^{M+1}\lambda_i r_i\le 0, \\ & A^T\mathbf{w}+\sum_{i=1}^{M+1}\boldsymbol{\rho}_i=\mathbf{0}, \\ & \|\mathbf{x}\|_1\le r_{M+1},\ \|\boldsymbol{\rho}_{M+1}\|_\infty\le\lambda_{M+1}\ \|\mathbf{x}^{(i)}\|_2\le r_i,\ \|\boldsymbol{\rho}_i\|_2\le\lambda_i,\ i=1,\dots M, \quad \boldsymbol{\lambda}\ge 0. \end{array} \tag{24}$$

We use a unified proof for the two problems. To this end, let P_i denote $\|\cdot\|_1$, $\|\cdot\|_2$ or $\frac{1}{2}\|\cdot\|_2^2$ of $\mathbf{x}^{(i)}$, where $\mathbf{x}^{(i)}$ is a sub-vector of $\mathbf{x}$. Further, we let

$$\begin{aligned} &[\tau]=J_1\cup J_2\cup J_3, \\ &P_{i_1}=\|\cdot\|_1,\ i_1\in J_1, \\ &P_{i_2}=\|\cdot\|_2,\ i_2\in J_2, \\ &P_{i_3}=\tfrac{1}{2}\|\cdot\|_2^2,\ i_3\in J_3. \end{aligned}$$

Lemma B.3. (11) *is equivalent to*

$$\begin{array}{ll} \min & L(\mathbf{x}) \\ \text{s.t.} & l(\mathbf{x})+\sum_{i=1}^{\tau}\frac{\frac{\lambda_i^k}{r_i^k}r_i^2+\frac{r_i^k}{\lambda_i^k}\lambda_i^2}{2}+l^*(-\sum_{i=1}^{\tau}\boldsymbol{\rho}_i)+\sum_{i\in J_3}s_i\le 0 \\ & \|\mathbf{x}\|_1\le r_i,\ i\in J_1,\ \|\mathbf{x}\|_2\le r_i,\ i\in J_2 \\ & \left\|\begin{pmatrix}\mathbf{x}\\ r_i-\frac{1}{2}\end{pmatrix}\right\|_2\le r_i+\tfrac{1}{2},\ i\in J_3 \\ & \|\boldsymbol{\rho}_i\|_\infty\le\lambda_i \text{ for } i\in J_1,\ \|\boldsymbol{\rho}_i\|_2\le\lambda_i \text{ for } i\in J_2 \\ & \left\|\begin{pmatrix}\sqrt{2}\boldsymbol{\rho}_i\\ s_i-\lambda_i\end{pmatrix}\right\|_2\le s_i+\lambda_i \quad \text{for } i\in J_3. \end{array} \tag{25}$$

Proof. We first simplify F defined by (7). Note that for $i\in J_1$

$$P_i^*(\mathbf{y})=\begin{cases}0, \text{ if } \|\mathbf{y}\|_\infty\le 1\\ \infty \text{ otherwise,}\end{cases}$$

for $i\in J_2$

$$P_i^*(\mathbf{y})=\begin{cases}0, \text{ if } \|\mathbf{y}\|_2\le 1\\ \infty \text{ otherwise,}\end{cases}$$

and for $i\in J_3$ $P_i(\mathbf{y})=\frac{1}{2}\|\mathbf{y}\|_2^2$. For $i\in J_3$, we introduce $\lambda_i P_i^*(\frac{\rho_i}{\lambda_i})\le s_i$. Then the constraints of (11) amount to

$$\begin{cases} l(\mathbf{x})+\sum_{i=1}^{\tau}\frac{\frac{\lambda_i^k}{r_i^k}r_i^2+\frac{r_i^k}{\lambda_i^k}\lambda_i^2}{2}+l^*(-\sum_{i=1}^{\tau}\boldsymbol{\rho}_i)\le 0 \\ \|\mathbf{x}\|_1\le r_i,\ i\in J_1,\ \|\mathbf{x}\|_2\le r_i,\ i\in J_2 \\ \frac{1}{2}\|\mathbf{x}\|_2^2\le r_i,\ i\in J_3 \\ \|\boldsymbol{\rho}_i\|_\infty\le\lambda_i,\ i\in J_1,\ \|\boldsymbol{\rho}_i\|_2\le\lambda_i,\ i\in J_2, \\ \frac{\|\boldsymbol{\rho}_i\|_2^2}{\lambda_i}\le 2s_i\ \ i\in J_3. \end{cases}$$

By $s_i\lambda_i=\frac{(s_i+\lambda_i)^2-(s_i-\lambda_i)^2}{4}$, $2r_i=(r_i+\frac{1}{2})^2-(r_i-\frac{1}{2})^2$ and taking square root, $\frac{\|\boldsymbol{\rho}_i\|_2^2}{\lambda_i}\le 2s_i$ and $\frac{1}{2}\|\mathbf{x}\|_2^2\le r_i$ are further equivalent to second-order cone constraints

$$\left\|\begin{pmatrix}\sqrt{2}\boldsymbol{\rho}_i\\ s_i-\lambda_i\end{pmatrix}\right\|_2\le s_i+\lambda_i,\ \left\|\begin{pmatrix}\mathbf{x}\\ r_i-\frac{1}{2}\end{pmatrix}\right\|_2\le r_i+\frac{1}{2}$$

which concludes the result. □

Many practical problems of interest choose the least square error as the loss function, i.e., $l(\mathbf{x})=\frac{1}{2}\|A\mathbf{x}-\mathbf{b}\|_2^2$. Particularly, we consider A^T to be of full row rank, i.e., the feature vectors of the data are linearly independent. In this case, (11) is further equivalent to a conically constrained convex problem.

Lemma B.4. *If $l(\mathbf{x}) = \frac{1}{2}\|A\mathbf{x}-\mathbf{b}\|_2^2$ with $A \in \mathbb{R}^{m\times n}$ being of full row rank, (11) can be further written as*

$$\begin{aligned}
\min\quad & L(\mathbf{x}) \\
\text{s.t.}\quad & \left\| \left(A\mathbf{x}-\mathbf{b}, \ldots, \sqrt{\tfrac{\lambda_i^k}{r_i^k}} r_i, \sqrt{\tfrac{r_i^k}{\lambda_i^k}}\lambda_i, \ldots, \mathbf{w}+\mathbf{b}, \tfrac{\sum_{i\in J_3} s_i}{\|\mathbf{b}\|_2} \right) \right\|_2 \le \|\mathbf{b}\|_2 - \tfrac{1}{\|\mathbf{b}\|_2} \sum_{i\in J_3} s_i \\
& A^T\mathbf{w} + \sum_{i=1}^{\tau} \boldsymbol{\rho}_i = \mathbf{0} \\
& \|\mathbf{x}\|_1 \le r_i,\ i\in J_1,\ \|\mathbf{x}\|_2 \le r_i,\ i\in J_2 \\
& \left\| \begin{pmatrix} \mathbf{x} \\ r_i - \frac{1}{2} \end{pmatrix} \right\|_2 \le r_i + \tfrac{1}{2},\ i\in J_3 \\
& \|\boldsymbol{\rho}_i\|_\infty \le \lambda_i,\ i\in J_1,\ \|\boldsymbol{\rho}_i\|_2 \le \lambda_i,\ i\in J_2 \\
& \left\| \begin{pmatrix} \sqrt{2}\boldsymbol{\rho}_i \\ s_i - \lambda_i \end{pmatrix} \right\|_2 \le s_i + \lambda_i,\ i\in J_3.
\end{aligned} \tag{26}$$

Proof. We first compute $l^*(\mathbf{y}) := \max_{\mathbf{x}}\ \mathbf{y}^T\mathbf{x} - \frac{1}{2}\|A\mathbf{x}-\mathbf{b}\|_2^2$, which is

$$l^*(\mathbf{y}) = \max_{\mathbf{x}}\ -\frac{1}{2}\mathbf{x}^T A^T A\mathbf{x} + \mathbf{x}^T\left(A^T\mathbf{b}+\mathbf{y}\right) - \frac{1}{2}\|\mathbf{b}\|^2.$$

By Example 9.1.1 in Boyd et al. (2004), if the above problem is solvable, $A^TA\mathbf{x}^* = A^T\mathbf{b}+\mathbf{y}$ is solvable with $\mathbf{x}^*$ being the optimal solution. This is equivalent to $\mathbf{y} = A^T\mathbf{w}$ for some $\mathbf{w}\in\mathbb{R}^m$ and then the full column rank of A yields $A\mathbf{x}^* = \mathbf{b}+\mathbf{w}$. Substituting it into $l^*(\mathbf{y})$, we have $l^*(\mathbf{y}) = \frac{1}{2}\|\mathbf{w}+\mathbf{b}\|_2^2 - \frac{1}{2}\|\mathbf{b}\|_2^2$, hence

$$l^*(\mathbf{y}) = \begin{cases} \frac{1}{2}\|\mathbf{w}+\mathbf{b}\|_2^2 - \frac{1}{2}\|\mathbf{b}\|_2^2 \text{ if } \mathbf{y} = A^T\mathbf{w} \\ \infty \text{ otherwise.} \end{cases}$$

Then the first constraint of (25) can be replaced with

$$\begin{cases} \|A\mathbf{x}-\mathbf{b}\|_2^2 + \sum_{i=1}^{\tau} \left\| \begin{pmatrix} \sqrt{\frac{\lambda_i^0}{r_i^0}} r_i \\ \sqrt{\frac{r_i^0}{\lambda_i^0}}\lambda_i \end{pmatrix} \right\|_2^2 + \|\mathbf{w}+\mathbf{b}\|_2^2 + 2\sum_{i\in J_3} s_i \le \|\mathbf{b}\|_2^2 \\ A^T\mathbf{w} + \sum_{i=1}^{\tau}\boldsymbol{\rho}_i = \mathbf{0}. \end{cases}$$

By using

$$\|\mathbf{b}\|_2^2 - 2\sum_{i\in J_3} s_i = \left(\|\mathbf{b}\|_2 - \frac{\sum_{i\in J_3} s_i}{\|\mathbf{b}\|_2} \right)^2 - \left(\frac{\sum_{i\in J_3} s_i}{\|\mathbf{b}\|_2} \right)^2$$

and taking the square root, we conclude the result. □

Now we are ready to give proofs for Propositions 3.2 and B.2.

Proof. Based on the above propositions, we give the expressions of the subproblems of the elastic net and sparse group lasso. Note that the expression of (8) is very similar to that of (11), we omit the augments for simplicity.

For the elastic net problem, $J_1 = \{1\}$ and $J_2 = \emptyset$ and $J_3 = \{2\}$. Then the conclusion follows from Lemma B.4 and introducing the variable t such that $t \ge L(\mathbf{x}) = \frac{1}{2}\|A_{val}\mathbf{x} - \mathbf{b}_{val}\|_2^2$, which is equivalent to $\left\| \begin{pmatrix} A_{val}\mathbf{x}-\mathbf{b}_{val} \\ t-\frac{1}{2} \end{pmatrix} \right\|_2 \le t+\frac{1}{2}$.

For the sparse group lasso problem, we let $J_2 = \{1,2,\ldots,M\}$ and $J_1 = \{M+1\}$ and the augments are the same as that of Proposition 3.2. □

Remark B.5. Without the linear independence of the data, we can still obtain a conic program for the subproblem but with one extra linear constraint. In fact, the linear independence, i.e., the full column rank of A^T, is only used in the proof of Lemma B.4 to yield $A\mathbf{x}^* = \mathbf{b}+\mathbf{w}$ from $A^TA\mathbf{x}^* = A^T\mathbf{b}+\mathbf{y}$ and $\mathbf{y} = A^T\mathbf{w}$. Without this condition, we could have

$$A\mathbf{x}^* = \mathbf{b}+\mathbf{w}+\mathbf{v} \text{ with } A^T\mathbf{v} = \mathbf{0},$$

and the arguments of conic program reformulation still go through.

B.2 Support Vector Machine

We now consider the support vector machine (SVM) problem

$$\begin{array}{ll}\min\limits_{\mathbf{w},c} & L(\mathbf{w},c)=\sum\limits_{j\in I_{val}}\max(1-b_j(\mathbf{w}^\top\mathbf{a}_j-c),0)\\ \text{s.t.} & \mathbf{w}\in\operatorname{argmin}\limits_{-\overline{\mathbf{w}}\le\mathbf{w}\le\overline{\mathbf{w}}}\sum\limits_{j\in I_{tr}}\max(1-b_j(\mathbf{w}^\top\mathbf{a}_j-c),0)+\frac{\lambda}{2}\|\mathbf{w}\|_2^2.\end{array}\tag{27}$$

Proposition B.6. *For the SVM problem, let $B_{tr}=\operatorname{diag}\{b_j,j\in I_{tr}\}$ denote the diagonal matrix whose diagnal elements consist of $\{b_i:i\in I_{tr}\}$ in sequence. Let $A_{tr}=(\mathbf{a}_i)_{i\in I_{tr}}$ denote the training data (without labels), $\mathbf{w},\mathbf{r}_2\in\mathbb{R}^p$ and $[r_2]_j$ denotes the j-th element of $\mathbf{r}_2$. Given $\overline{\mathbf{w}}^0,\mathbf{r}_2^0>0$, (11) for problem (27) can be reformulated as*

$$\begin{array}{ll}\min & L(\mathbf{w},c)=\sum\limits_{j\in I_{val}}\max(1-b_j(\mathbf{w}^\top\mathbf{a}_j-c),0)\\ \text{s.t.} & \sum\limits_{j\in I_{tr}}\max(1-b_j(\mathbf{w}^\top\mathbf{a}_j-c),0)+\frac{1}{2}(\frac{\lambda^0}{r_1^0}r_1^2+\frac{r_1^0}{\lambda^0}\lambda^2)+\frac{1}{2}\sum\limits_{j=1}^{p}\left(\frac{\overline{w}_j^0}{[r_2^0]_j}[r_2]_j{}^2+\frac{[r_2^0]_j}{\overline{w}_j^0}\overline{w}_j^2\right)+s-\mathbf{1}^T\mathbf{v}\le 0,\\ & \frac{1}{2}\|\mathbf{w}\|^2\le r_1,\\ & \left\|\begin{pmatrix}\sqrt{2}\rho\\ \lambda-s\end{pmatrix}\right\|_2\le\lambda+s,\\ & \begin{pmatrix}A_{tr}^T\\ \mathbf{1}^T\end{pmatrix}B_{tr}\mathbf{v}+\begin{pmatrix}-\boldsymbol{\rho}\\ 0\end{pmatrix}+\begin{pmatrix}\boldsymbol{\alpha}_2-\boldsymbol{\alpha}_1\\ 0\end{pmatrix}=0,\\ & \boldsymbol{\alpha}_1+\boldsymbol{\alpha}_2=\mathbf{r}_2,\\ & 0\le\mathbf{v}\le 1,\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2\ge 0,\\ & -\overline{\mathbf{w}}\le\mathbf{w}\le\overline{\mathbf{w}}.\end{array}\tag{28}$$

Moreover, (8) for SVM is

$$\begin{array}{ll}\min & L(\mathbf{w},c)=\sum\limits_{j\in I_{val}}\max(1-b_j(\mathbf{w}^\top\mathbf{a}_j-c),0),\\ \text{s.t.} & \sum\limits_{j\in I_{tr}}\max(1-b_j(\mathbf{w}^\top\mathbf{a}_j-c),0)+r_1\lambda+\mathbf{r}_2^\top\overline{\mathbf{w}}+s-\mathbf{1}^T\mathbf{v}\le 0,\\ & \frac{1}{2}\|\mathbf{w}\|^2\le r_1,\\ & \left\|\begin{pmatrix}\sqrt{2}\boldsymbol{\rho}\\ \lambda-s\end{pmatrix}\right\|_2\le\lambda+s,\\ & \begin{pmatrix}A_{tr}^T\\ \mathbf{1}^T\end{pmatrix}B_{tr}\mathbf{v}+\begin{pmatrix}-\boldsymbol{\rho}\\ 0\end{pmatrix}+\begin{pmatrix}\boldsymbol{\alpha}_2-\boldsymbol{\alpha}_1\\ 0\end{pmatrix}=0,\\ & \boldsymbol{\alpha}_1+\boldsymbol{\alpha}_2=\mathbf{r}_2,\\ & 0\le\mathbf{v}\le 1,\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2\ge 0,\\ & -\overline{\mathbf{w}}\le\mathbf{w}\le\overline{\mathbf{w}}.\end{array}\tag{29}$$

We remark that (28) is also equivalent to a conic program by the same augments of Lemma B.4 and the proofs are omitted for the sake of brevity.

Proof. The lower-level problem can be written as

$$\begin{array}{ll}\min\limits_{\mathbf{w},c,\mathbf{z}} & l(\mathbf{w},c)+\lambda P(\mathbf{z})\\ \text{s.t.} & \mathbf{w}=\mathbf{z},\quad g_1(\mathbf{w},c)\le 0,\ g_2(\mathbf{w},c)\le 0,\end{array}$$

where $l(\mathbf{w},c)=\sum\limits_{j\in I_{tr}}\max(1-b_j(\mathbf{w}^\top a_j-c),0)$, $P=\frac{1}{2}\|\cdot\|_2^2$, $g_1(\mathbf{w},c)=\mathbf{w}-\overline{\mathbf{w}}$ and $g_2(\mathbf{w},c)=-\mathbf{w}-\overline{\mathbf{w}}$. The strong duality holds since the constraints are linear, then the above problem is further equivalent to

$$\begin{array}{lll}\max\limits_{\rho,\boldsymbol{\alpha}_1\ge 0,\boldsymbol{\alpha}_2\ge 0} & \min\limits_{\mathbf{w},c,\mathbf{z}} & l(\mathbf{w},c)+\lambda P(\mathbf{z})+\boldsymbol{\rho}^T(\mathbf{w}-\mathbf{z})+\boldsymbol{\alpha}_1^Tg_1(\mathbf{w},c)+\boldsymbol{\alpha}_2^Tg_2(\mathbf{w},c)\\ =\max\limits_{\boldsymbol{\rho},\boldsymbol{\alpha}_1\ge 0,\boldsymbol{\alpha}_2\ge 0} & -\max\limits_{\mathbf{w},c,\mathbf{z}} & -\boldsymbol{\rho}^\top\mathbf{w}-l(\mathbf{w},c)+\rho^\top\mathbf{z}-\lambda P(\mathbf{z})-(\boldsymbol{\alpha}_1^Tg_1(\mathbf{w},c)+\boldsymbol{\alpha}_2^Tg_2(\mathbf{w},c))\\ =\max\limits_{\boldsymbol{\rho},\boldsymbol{\alpha}_1\ge 0,\boldsymbol{\alpha}_2\ge 0} & & -l_g^*(-\boldsymbol{\rho},0)-\lambda P^*(\frac{\boldsymbol{\rho}}{\lambda})\\ =-\min\limits_{\boldsymbol{\rho},\boldsymbol{\alpha}_1\ge 0,\boldsymbol{\alpha}_2\ge 0} & & l_g^*(-\boldsymbol{\rho},0)+\lambda P^*(\frac{\boldsymbol{\rho}}{\lambda}),\end{array}$$

where $l_g(\mathbf{w},c) = l(\mathbf{w},c) + \boldsymbol{\alpha}_1^T g_1(\mathbf{w},c) + \boldsymbol{\alpha}_2^T g_2(\mathbf{w},c)$. Then problem (8) for SVM is equivalent to

$$\begin{array}{ll} \min\limits_{\mathbf{w},c} & L(\mathbf{w},c) \\ \text{s.t.} & l(\mathbf{w},c) + \lambda P(\mathbf{w}) \leq -l_g^*(-\boldsymbol{\rho}) - \lambda P^*(\frac{\rho}{\lambda}). \end{array} \tag{30}$$

Specifically, by Table 1, we have

$$l_g(\mathbf{w},c) = \sum_{j\in I_{tr}} \max(1 - b_j(\mathbf{w}^\top \mathbf{a}_j - c), 0) + \boldsymbol{\alpha}_1^T(\mathbf{w} - \overline{\mathbf{w}}) + \boldsymbol{\alpha}_2^T(-\mathbf{w} - \overline{\mathbf{w}}).$$

We calculate the conjugate function as follows

$$\begin{array}{rll} l_g^*(\mathbf{y},t) & = \max\limits_{\mathbf{w},c} & \{(\mathbf{y}^\top,t)(\mathbf{w},c)^\top - \sum\limits_{j\in I_{tr}} \max(1 - b_j(\mathbf{w}^\top a_j - c), 0) - \boldsymbol{\alpha}_1^T(\mathbf{w} - \overline{\mathbf{w}}) - \boldsymbol{\alpha}_2^T(-\mathbf{w} - \overline{\mathbf{w}})\} \\ & = \max\limits_{\mathbf{w},c,\mathbf{u}} & \{(\mathbf{y}^\top,t)(\mathbf{w},c)^\top - \sum\limits_{j\in I_{tr}} u_j - \boldsymbol{\alpha}_1^T(\mathbf{w} - \overline{\mathbf{w}}) - \boldsymbol{\alpha}_2^T(-\mathbf{w} - \overline{\mathbf{w}})\} \\ & \text{s.t.} & u_j \geq 1 - b_j(\mathbf{w}^\top a_j - c), u_j \geq 0, j \in I_{tr}. \end{array}$$

Hence we have

$$\begin{array}{rll} l_g^*(\mathbf{y},t) & = \max\limits_{\mathbf{w},c,\mathbf{u}} & \{(\mathbf{y}^\top,t)(\mathbf{w},c)^\top - \mathbf{1}^\top\mathbf{u} + (\boldsymbol{\alpha}_1^T + \boldsymbol{\alpha}_2^T)\overline{\mathbf{w}} - (\boldsymbol{\alpha}_1^T - \boldsymbol{\alpha}_2^T)\mathbf{w}\} \\ & \text{s.t.} & u_j \geq 1 - b_j(\mathbf{w}^\top \mathbf{a}_j - c), z_j \geq 0, j \in I_{tr}. \end{array} \tag{31}$$

Note that (31) is indeed a linear program and we simplify it by using duality. Let $L_{\boldsymbol{\gamma}}$ denote the Lagrange function of (31) and $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2$ denote the multipliers. Then

$$L_\gamma(\mathbf{w},c,\mathbf{u};\boldsymbol{\gamma}_1,\boldsymbol{\gamma}_2) = (\mathbf{y},t)^T(\mathbf{w},c) - \mathbf{1}^\top\mathbf{z} + (\boldsymbol{\alpha}_1+\boldsymbol{\alpha}_2)^T\overline{\mathbf{w}} - (\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2)^T\mathbf{w} + \boldsymbol{\gamma}_1^\top\mathbf{u} + \boldsymbol{\gamma}_2^\top(\mathbf{u} - \mathbf{1} + B_{tr}A_{tr}^\top\mathbf{w} - cB_{tr}\mathbf{1}). \tag{32}$$

where $B_{tr} = \text{diag}\{b_j, j \in I_{tr}\}$ is a diagonal matrix whose diagnal elements consist of $\{[b_{tr}]_i : i \in I_{tr}\}$ in sequence. By calculating the minimum value of the Lagrangian over $(\mathbf{w}, c, \mathbf{u})$, we obtain the dual function as follows.

$$\begin{array}{rll} l_g^*(\mathbf{y},t) & = \min\limits_{\mathbf{y},t} & -\boldsymbol{\gamma}_2^\top\mathbf{1} + (\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2)^\top\overline{\mathbf{w}} \\ & \text{s.t.} & \boldsymbol{\gamma}_1 + \boldsymbol{\gamma}_2 - \mathbf{1} = 0, \\ & & \begin{pmatrix}\mathbf{y}\\ t\end{pmatrix} + \begin{pmatrix}A_{tr}^\top \\ \mathbf{1}^T\end{pmatrix} B_{tr}\boldsymbol{\gamma}_2 + \begin{pmatrix}\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_1 \\ 0\end{pmatrix} = 0. \end{array} \tag{33}$$

By introducing $\mathbf{r}_2 = \boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2$ and recalling $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \geq 0$, we conclude that (33) is equivalent to

$$\begin{array}{rll} l_g^*(\mathbf{y},t) & = \min\limits_{\mathbf{y},t} & -\boldsymbol{\gamma}_2^\top\mathbf{1} + \mathbf{r}_2^\top\overline{\mathbf{w}} \\ & \text{s.t.} & \boldsymbol{\gamma}_1 + \boldsymbol{\gamma}_2 - \mathbf{1} = 0, \\ & & \boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2 = \mathbf{r}_2, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \geq 0, \\ & & \begin{pmatrix}\mathbf{y}\\ t\end{pmatrix} + \begin{pmatrix}A_{tr}^\top \\ \mathbf{1}^T\end{pmatrix} B_{tr}\boldsymbol{\gamma}_2 + \begin{pmatrix}\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_1 \\ 0\end{pmatrix} = 0. \end{array} \tag{34}$$

Note that in (30), $\lambda P^*(\frac{\boldsymbol{\rho}}{\lambda}) = \frac{\|\boldsymbol{\rho}\|_2^2}{2\lambda}$. We introduce $\frac{1}{2}\|\mathbf{w}\|_2^2 \leq r_1$, $\frac{\|\boldsymbol{\rho}\|_2^2}{2\lambda} \leq s$. By combining (30) (33) and using similar augments of Lemma B.3, we conclude that (8) is equivalent to (29) for SVM. Using the expression $m_1(\lambda, r_1; \lambda^0, r_1^0) = \frac{1}{2}\left(\frac{\lambda^0}{r_1^0}r_1^2 + \frac{r_1^0}{\lambda^0}\lambda^2\right)$ and $m_2(\overline{\mathbf{w}}, \mathbf{r}_2; \overline{\mathbf{w}}^0, \mathbf{r}_2^0) = \frac{1}{2}\sum\limits_{j=1}^p\left(\frac{\overline{w}_j^0}{[r_2^0]_j}[r_2]_j{}^2 + \frac{[r_2^0]_j}{\overline{w}_j^0}\overline{w}_j^2\right)$, (28) follows. □

B.3 Low-rank Matrix Completion

We now discuss the low-rank matrix completion model and summarise problem as follows

$$\begin{array}{ll} \min\limits_{\boldsymbol{\lambda}\in\mathbb{R}_+^{2G+1}} & \frac{1}{2}\|M_{val} - X_{val}\boldsymbol{\theta}\mathbf{1}^\mathbf{T} - (Z_{val}\boldsymbol{\beta}\mathbf{1}^T)^T - \Gamma\|_F^2 \\ \text{s.t.} & (\boldsymbol{\beta},\Gamma) \in \text{argmin}_{\boldsymbol{\beta},\Gamma}\, \frac{1}{2}\|M_{tr} - X_{tr}\boldsymbol{\theta}\mathbf{1}^\mathbf{T} - (Z_{tr}\boldsymbol{\beta}\mathbf{1}^T)^T - \Gamma\|_F^2 \\ & + \lambda_0\|\Gamma\|_* + \sum\limits_{g=1}^G \lambda_g\|\boldsymbol{\theta}^{(g)}\|_2 + \sum\limits_{g=1}^G \|\boldsymbol{\beta}^{(g)}\|_2 \end{array} \tag{35}$$

where $M_{val} = \{M_{ij}\}_{(i,j)\in\Omega_{val}}, M_{tr} = \{M_{ij}\}_{(i,j)\in\Omega_{tr}}$, $X_{val} = (\mathbf{x}_i)_{i\in I_{val}}, X_{tr} = (\mathbf{x}_i)_{i\in I_{tr}}$ and $Z_{val} = (\mathbf{z}_j)_{j\in I_{val}}, Z_{tr} = (\mathbf{z}_j)_{j\in I_{tr}}$

Proposition B.7. *We denote the spectral norm of W as $\|W\|_p$ and corresponding matrice as above. (11) for problem (35) can be reformulated as*

$$\begin{aligned}
\min_{\boldsymbol{\lambda}\in\mathbb{R}_+^{2G+1}} \quad & \tfrac{1}{2}\|M_{val} - X_{val}\boldsymbol{\theta}\mathbf{1}^{\mathbf{T}} - (Z_{val}\boldsymbol{\beta}\boldsymbol{1}^T)^T - \Gamma\|_F^2 \\
\text{s.t.} \quad & \tfrac{1}{2}\|M_{tr} - X_{tr}\boldsymbol{\theta}\mathbf{1}^{\mathbf{T}} - (Z_{tr}\boldsymbol{\beta}\boldsymbol{1}^T)^T - \Gamma\|_F^2 + \mathrm{tr}(M_{tr}^T W) + \tfrac{1}{2}\|W\|_F^2 + \tfrac{1}{2}\sum_{g=0}^{2G}\tfrac{\lambda_g^0}{r_g^0}r_g^2 + \tfrac{r_g^0}{\lambda_g^0}\lambda_g^2 \le 0, \\
& -\boldsymbol{\rho}_1 + X_{tr}^T W\mathbf{1} = \mathbf{0}, \\
& -\boldsymbol{\rho}_2 + Z_{tr}^T W^T\mathbf{1} = \mathbf{0}, \\
& \|\Gamma\|_* \le r_0, \\
& \left\|\boldsymbol{\theta}^{(g)}\right\|_2 \le r_g,\ g = 1,...,G, \\
& \left\|\boldsymbol{\beta}^{(g)}\right\|_2 \le r_{g+G},\ g = 1,...,G, \\
& \left\|\boldsymbol{\rho}_1^{(g)}\right\|_2 \le \lambda_g,\ g = 1,...,G, \\
& \left\|\boldsymbol{\rho}_2^{(g)}\right\|_2 \le \lambda_{g+G},\ g = 1,...,G, \\
& \|W\|_p \le \lambda_0.
\end{aligned} \tag{36}$$

Moreover, (8) for low-rank matrix completion is

$$\begin{aligned}
\min_{\boldsymbol{\lambda}\in\mathbb{R}_+^{2G+1}} \quad & \tfrac{1}{2}\|M_{val} - X_{val}\boldsymbol{\theta}\mathbf{1}^{\mathbf{T}} - (Z_{val}\boldsymbol{\beta}\boldsymbol{1}^T)^T - \Gamma\|_F^2 \\
\text{s.t.} \quad & \tfrac{1}{2}\|M_{tr} - X_{tr}\boldsymbol{\theta}\mathbf{1}^{\mathbf{T}} - (Z_{tr}\boldsymbol{\beta}\boldsymbol{1}^T)^T - \Gamma\|_F^2 + \mathrm{tr}(M_{tr}^T W) + \tfrac{1}{2}\|W\|_F^2 + \sum_{g=0}^{2G}\lambda_g r_g \le 0, \\
& -\boldsymbol{\rho}_1 + X_{tr}^T W\mathbf{1} = \mathbf{0} \\
& -\boldsymbol{\rho}_2 + Z_{tr}^T W^T\mathbf{1} = \mathbf{0} \\
& \|\Gamma\|_* \le r_0, \\
& \left\|\boldsymbol{\theta}^{(g)}\right\|_2 \le r_g,\ g = 1,...,G, \\
& \left\|\boldsymbol{\beta}^{(g)}\right\|_2 \le r_{g+G},\ g = 1,...,G, \\
& \left\|\boldsymbol{\rho}_1^{(g)}\right\|_2 \le \lambda_g,\ g = 1,...,G, \\
& \left\|\boldsymbol{\rho}_2^{(g)}\right\|_2 \le \lambda_{g+G},\ g = 1,...,G, \\
& \|W\|_p \le \lambda_0.
\end{aligned} \tag{37}$$

Proof. We define

$$l(\boldsymbol{\theta},\boldsymbol{\beta},\Gamma) = \frac{1}{2}\|M_{tr} - X_{tr}\boldsymbol{\theta}\mathbf{1}^{\mathbf{T}} - (Z_{tr}\boldsymbol{\beta}\mathbf{1}^T)^T - \Gamma\|_F^2,$$

and compute its conjugate function

$$l^*(\mathbf{u},\mathbf{v},W) = \max_{\boldsymbol{\theta},\boldsymbol{\beta},\Gamma} g(\boldsymbol{\theta},\boldsymbol{\beta},\Gamma) := \max \boldsymbol{\theta}^T\mathbf{u} + \boldsymbol{\beta}^T\mathbf{v} + \mathrm{tr}(\Gamma^T W) - l(\boldsymbol{\theta},\boldsymbol{\beta},\Gamma).$$

By first order condition, for optimal $(\boldsymbol{\theta}^*,\boldsymbol{\beta}^*,\Gamma^*)$, it holds that

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta},\boldsymbol{\beta},\Gamma) &= 0, \\
\nabla_{\boldsymbol{\beta}} g(\boldsymbol{\theta},\boldsymbol{\beta},\Gamma) &= 0, \\
\nabla_{\Gamma} g(\boldsymbol{\theta},\boldsymbol{\beta},\Gamma) &= 0,
\end{aligned}$$

which are equivalent to

$$\mathbf{u} + X_{tr}^T(M_{tr} - X_{tr}\boldsymbol{\theta}\mathbf{1}^T - (Z_{tr}\boldsymbol{\beta}\mathbf{1}^T)^T - \Gamma)\mathbf{1} = \mathbf{0}, \tag{38}$$

$$\mathbf{v} + Z_{tr}^T(M_{tr} - X_{tr}\boldsymbol{\theta}\mathbf{1}^T - (Z_{tr}\boldsymbol{\beta}\mathbf{1}^T)^T - \Gamma)^T\mathbf{1} = \mathbf{0}, \tag{39}$$

$$W - (M_{tr} - X_{tr}\boldsymbol{\theta}\mathbf{1}^T - (Z_{tr}\boldsymbol{\beta}\mathbf{1}^T)^T - \Gamma) = \mathbf{0}. \tag{40}$$

Substituting (40) into (39) and (38), we obtain

$$\mathbf{u} - X_{tr}^T W\mathbf{1} = \mathbf{0}, \quad \mathbf{v} - Z_{tr}^T W^T\mathbf{1} = \mathbf{0}. \tag{41}$$

Hence $l^*(\mathbf{u}, \mathbf{v}, W) = +\infty$ if (41) is not satisfied. We choose $\boldsymbol{\theta}^* = \mathbf{0}, \boldsymbol{\beta}^* = \mathbf{0}, \Gamma^* = M_{tr} + W$ and obtain that

$$l^*(\mathbf{u}, \mathbf{v}, W) = \begin{cases} \mathrm{tr}(M_{tr}^T W) + \frac{1}{2}\|W\|_F^2 \text{ if (41) holds,} \\ +\infty \text{ otherwise.} \end{cases} \tag{42}$$

By combining (42) and using similar augments of Lemma B.3, we conclude that (35) is equivalent to for low-rank matrix completion. Using the expression $m(\lambda_g, r_g; \lambda_g^0, r_g^0) = \frac{1}{2}\left(\frac{\lambda_g^0}{r_g^0} r_g^2 + \frac{r_g^0}{\lambda_g^0}\lambda_g^2\right)$, (36) follows.

□

C Detials for Experiments and Data

C.1 Elastic Net

The generation of feature matrix $A \in \mathbb{R}^{n\times p}$ and response vector $\mathbf{b} \in \mathbb{R}^n$ follows Feng and Simon (2018), where the column vectors $\mathbf{a}_i \in \mathbb{R}^p : i \in I_{tr} \setminus I_{val}$ satisfy the marginal distribution $N(\mathbf{0}, \mathbf{I})$, and the correlation matrix between column vectors satisfies $cor(a_{ij}, a_{ik}) = 0.5^{|j-k|}$. The feature matrix is full rank and satisfies the conditions in Lemma B.4. Next, we generate a random vector $\boldsymbol{\beta} \in \mathbb{R}^p$ with 15 non-zero elements, where each element β_i is either 0 or 1. The response vector $\mathbf{b}$ is obtained by applying the feature matrix to the random vector and adding a certain amount of noise, i.e., $\mathbf{b} = \mathbf{A}\boldsymbol{\beta} + \sigma\boldsymbol{\epsilon}$, where we set the signal-to-noise ratio to $\sigma = 2$ and the noise $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_n)$. Random search is implemented using 100 uniformly random samples. The variable space of TPE is set to a uniform distribution on $[-5, 2]$ for both u_1 and u_2. We follow Gao et al. (2022) and use the same parameter settings and stopping criteria to implement the VF-iDCA algorithm. For LDMMA algorithm, we set the initial point to $\boldsymbol{\lambda}^0 = (0.01, 0.01)$. For the ϵ-perturbation problem (13), we set $\epsilon = 0.01$.

C.2 Sparse Group Lasso

The generation of the feature matrix $A \in \mathbb{R}^{n\times p}$ and the response vector $\mathbf{b} \in \mathbb{R}^n$ follows Feng and Simon (2018). The generated dataset includes n training samples, $n/3$ validation samples, and 100 fixed testing samples. The observation matrix $\mathbf{A}$ satisfies that each column vector $\mathbf{a}_i$ follows the standard normal distribution. The random vector $\boldsymbol{\beta} = [\boldsymbol{\beta}^{(1)}, \boldsymbol{\beta}^{(2)}, \boldsymbol{\beta}^{(3)}] \in \mathbb{R}^p$, where $\boldsymbol{\beta}^{(i)} = (1, 2, 3, 4, 5, 0, ..., 0)$. The response vector $\mathbf{b}$ is generated by applying the feature matrix to the random vector and adding some noise. Specifically, $b_i = \boldsymbol{\beta}^T \mathbf{a}_i + \sigma\epsilon_i$ and $\mathbf{b} = (b_1, b_2, ..., b_n)$. Like the elastic net model, we set the signal-to-noise ratio to $\sigma = 2$ and the noise $\boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_n) \sim N(\mathbf{0}, \mathbf{I})$. For the experiments with four different data sizes, the algorithm details of VF-iDCA are the same as Gao et al. (2022), including parameter tuning. For LDMMA, the initial value of the iteration is set to $\boldsymbol{\lambda}^0 = (0.1, ..., 0.1)$, and $\epsilon = 1$ is also set. It is worth noting that if ϵ is too small, the feasible domain of problem (13) is insufficient to complete the iteration. This premature termination will result in an abnormally low validation error and a larger test error. In machine learning, we call it overfitting, which is usually caused by poor generalization performance of the model. The occurrence of overfitting indicates that the model only has a good learning effect on the training and validation data but has no practical value for unlearned test data and more extensive data. Overall, in this series of experiments, we need to choose an appropriate value of ϵ, which can avoid overfitting and prevent the solution of problem (13) from deviating too much from the solution of the original problem (1).

C.3 Support Vector Machine

We fetch the datasets with libSVM toolbox and obtain the corresponding observation matrix and label vector of all datasets. Each dataset is divided into two seperate parts: a cross-validation training set Ω containing $3\lfloor N/6 \rfloor$ samples, and a test set Ω_{test} containing the remaining samples. Based on this devision, we partition the entire training set into multiple equal parts and iteratively use one part as the validation set and the remaining parts as the training set to solve the SVM problem. In the experiment, we performed 3-fold and 6-fold cross-validation on the training and validation sets for each of the six datasets to optimize hyperparameters.

Finally, we use the obtained hyperparameters and corresponding model to compute the error on the validation set. We repeat this process for each part to reduce the impact of data variability on the model. However, in the process of solving the SVM problem, cross-validation is involved, and the obtained hyperparameters satisfy the

minimization of the lower-level function of the SVM problem. Therefore, we need to use the hyperparameters to solve the upper-level problem to obtain the corresponding validation and test errors. We randomly divide the cross-validation training set Ω into K mutually exclusive subsets $\{\Omega^k_{val}\}_{k=1}^K$, each of which will be used as the validation set. The remaining parts will be used as the training set $\Omega^k_{tr} = \Omega \setminus \Omega^k_{val}$. We define the loss function on the validation set in the cross-validation process as:

$$\Theta_{val}(\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^K, \mathbf{c}) := \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\Omega^k_{val}|} \sum_{j \in \Omega^k_{val}} \max(1 - b_j(\mathbf{a}_j^\top \mathbf{w}^k - c^k), 0), \tag{43}$$

The primal problem of the support vector machine (27) is then transformed into the following bilevel program (Kunapuli et al., 2008):

$$\begin{aligned} \min_{\mathbf{w},c} \quad & \Theta_{val}(\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^K, \mathbf{c}) \\ \text{s.t.} \quad & \lambda > 0, \bar{\mathbf{w}}_{lb} \le \bar{\mathbf{w}} \le \bar{\mathbf{w}}_{ub} \\ & (\mathbf{w}^k, c^k) \in \operatorname*{arg\,min}_{-\overline{\mathbf{w}} \le \mathbf{w} \le \overline{\mathbf{w}}} \left\{ \sum_{j \in \Omega^k_{tr}} \max(1 - b_j(\mathbf{a}_j^\top \mathbf{w} - c), 0) + \tfrac{\lambda}{2}\|\mathbf{w}\|_2^2 \right\}, \\ & k = 1, 2, \ldots, K. \end{aligned} \tag{44}$$

where $\mathbf{c} = (c^1, c^2, \ldots, c^K)$, $c^1, c^2, \ldots, c^K$ and $\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^K$ are K parallel copies of c and $\mathbf{w}$. $\bar{\mathbf{w}}lb$ and $\bar{\mathbf{w}}lb$ are the upper and lower bounds of $\bar{\mathbf{w}}$, respectively. We can define a loss function on the test set analogous to (43):

$$\begin{aligned} & \Theta_{tr}(\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^K, \mathbf{c}) \\ & := \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\Omega^k_{tr}|} \sum_{j \in \Omega^k_{tr}} \max(1 - b_j(\mathbf{a}_j^\top \mathbf{w}^k - c^k), 0), \end{aligned} \tag{45}$$

Correspondingly, the subproblem (29) to be solved is transformed into:

$$\begin{aligned} \min_{\lambda, \bar{\mathbf{w}}, \mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^K, c} \quad & \Theta_{val}(\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^K, \mathbf{c}), \\ \text{s.t.} \quad & \Theta_{tr}(\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^K, \mathbf{c}) + r_1\lambda + \mathbf{r}_2^\top \overline{\mathbf{w}} + s - \mathbf{1}^T \mathbf{v} \le 0, \\ & \tfrac{1}{2}\|\mathbf{w}^k\|^2 \le r_1, k = 1, 2, \ldots, K \\ & \left\| \begin{matrix} \sqrt{2}\boldsymbol{\rho} \\ \lambda - s \end{matrix} \right\|_2 \le \lambda + s, \\ & \begin{pmatrix} A^{k\,T}_{tr} \\ \mathbf{1}^T \end{pmatrix} B^k_{tr}\mathbf{v} + \begin{pmatrix} -\boldsymbol{\rho} \\ 0 \end{pmatrix} + \begin{pmatrix} \boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_1 \\ 0 \end{pmatrix} = 0, k = 1, 2, ..., K \\ & \boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2 = \mathbf{r}_2, \\ & 0 \le \mathbf{v} \le 1, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \ge 0, \\ & -\overline{\mathbf{w}} \le \mathbf{w}^k \le \overline{\mathbf{w}}, k = 1, 2, \ldots, K \\ & \bar{\mathbf{w}}_{lb} \le \bar{\mathbf{w}} \le \bar{\mathbf{w}}_{ub} \end{aligned} \tag{46}$$

Finally, we substitute the optimal solutions $\bar{\mathbf{w}}, \lambda$ obtained from the above problem into the following problem and solve it again to obtain the optimal $(\mathbf{w}, c)$,

$$(\mathbf{w}, c) \in \operatorname*{arg\,min}_{-\overline{\mathbf{w}} \le \mathbf{w} \le \overline{\mathbf{w}}} \left\{ \sum_{j \in \Omega} \max(1 - b_j(\mathbf{a}_j^\top \mathbf{w} - c), 0) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 \right\}.$$

We use MOSEK solver to handle the 2-norm term in the objective function of the upper-level problem, which is convex and smooth. We also conduct VF-iDCA and other methods according to the setting in Gao et al. (2022). For LDMMA, we set the initial point of the iteration to $\boldsymbol{\lambda}^0 = (0.1, \ldots, 0.1)$ and parameters $\bar{\mathbf{w}}_{lb} = (10^{-6}, \ldots, 10^{-6})$, $\bar{\mathbf{w}}_{ub} = (10, \ldots, 10)$ for the lower and upper bounds of $\bar{\mathbf{w}}$, respectively. We choose $\epsilon = 1$ for 3-fold cross-validation and $\epsilon = 5$ for 6-fold cross-validation to ensure the primal and dual feasibility of the subproblems in each iteration, which is crucial for MOSEK and prevents overfitting.

C.4 Elastic Net with High Dimensional Datasets

Compared with Section C.1, we only replace the synthetic datasets with real datasets. The gisette dataset comprises 5000 features and 6000 samples, whereas the sensit dataset encompasses 78823 features. For dataset

partition, we extract 50, 25 examples as training set and 50, 25 examples as validation set, respectively. We set the initial point as $\boldsymbol{\lambda}^0 = (0.01, 0.01)$ and perturbation parameter $\epsilon = 1$ for our algorithm. Meanwhile, we also conduct VF-iDCA and other methods according to the setting in Gao et al. (2022).

Table 6: Support Vector Machine problems with 3-fold and 6-fold cross-validation on three datasets, where the number of features p and samples $|\Omega|, |\Omega_{test}|$ are displayed together with dataset names.

Dataset	**Methods**	**3-fold**			**6-fold**		
		Times(s)	**Val. Err.**	**Test Err.**	**Times(s)**	**Val. Err.**	**Test Err.**
liver-disorders-scale $p = 5$ $\|\Omega\| = 72$ $\|\Omega_{test}\| = 73$	Grid	0.74 ± 0.01	0.65 ± 0.08	0.32 ± 0.07	1.16 ± 0.02	0.61 ± 0.08	0.32 ± 0.06
	Random	0.75 ± 0.02	0.63 ± 0.07	0.32 ± 0.05	1.16 ± 0.04	0.59 ± 0.06	0.32 ± 0.05
	TPE	0.68 ± 0.55	0.65 ± 0.08	0.32 ± 0.07	2.26 ± 1.67	0.62 ± 0.06	0.32 ± 0.06
	VF-iDCA	0.13 ± 0.03	0.52 ± 0.07	0.27 ± 0.04	0.27 ± 0.03	0.40 ± 0.05	0.30 ± 0.04
	LDMMA	0.08 ± 0.01	0.46 ± 0.08	0.23 ± 0.10	0.15 ± 0.04	0.19 ± 0.08	0.24 ± 0.08
diabetes-scale $p = 8$ $\|\Omega\| = 384$ $\|\Omega_{test}\| = 384$	Grid	3.17 ± 0.08	0.55 ± 0.03	0.19 ± 0.03	6.22 ± 0.21	0.54 ± 0.03	0.33 ± 0.04
	Random	3.47 ± 0.14	0.56 ± 0.03	0.32 ± 0.05	7.18 ± 0.30	0.55 ± 0.04	0.30 ± 0.05
	TPE	10.21 ± 6.68	0.55 ± 0.04	0.29 ± 0.06	76.67 ± 36.39	0.54 ± 0.03	0.34 ± 0.06
	VF-iDCA	0.28 ± 0.04	0.48 ± 0.03	0.23 ± 0.01	0.65 ± 0.03	0.43 ± 0.03	0.23 ± 0.02
	LDMMA	0.22 ± 0.03	0.49 ± 0.02	0.19 ± 0.01	0.55 ± 0.10	0.39 ± 0.05	0.20 ± 0.02
breast-cancer-scale $p = 14$ $\|\Omega\| = 336$ $\|\Omega_{test}\| = 347$	Grid	3.32 ± 0.09	0.08 ± 0.01	0.16 ± 0.08	6.32 ± 0.11	0.08 ± 0.01	0.15 ± 0.12
	Random	3.69 ± 0.07	0.09 ± 0.01	0.08 ± 0.08	7.20 ± 0.12	0.09 ± 0.02	0.10 ± 0.11
	TPE	17.88 ± 10.05	0.09 ± 0.01	0.10 ± 0.11	34.66 ± 20.57	0.09 ± 0.01	0.18 ± 0.13
	VF-iDCA	0.24 ± 0.04	0.09 ± 0.01	0.04 ± 0.01	0.57 ± 0.12	0.08 ± 0.01	0.03 ± 0.01
	LDMMA	0.12 ± 0.01	0.08 ± 0.01	0.03 ± 0.01	0.42 ± 0.17	0.08 ± 0.01	0.02 ± 0.01
sonar $p = 60$ $\|\Omega\| = 102$ $\|\Omega_{test}\| = 106$	Grid	10.08 ± 0.33	0.59 ± 0.10	0.41 ± 0.14	20.88 ± 0.61	0.63 ± 0.06	0.49 ± 0.12
	Random	10.30 ± 0.18	0.55 ± 0.07	0.31 ± 0.08	20.56 ± 0.31	0.58 ± 0.03	0.41 ± 0.10
	TPE	42.80 ± 13.95	0.64 ± 0.13	0.45 ± 0.11	189.82 ± 19.80	0.70 ± 0.06	0.53 ± 0.07
	VF-iDCA	1.32 ± 0.23	0.03 ± 0.02	0.25 ± 0.04	3.03 ± 0.09	0.00 ± 0.00	0.24 ± 0.04
	LDMMA	0.82 ± 0.15	0.17 ± 0.02	0.25 ± 0.04	2.38 ± 0.19	0.00 ± 0.00	0.22 ± 0.02
a1a $p = 123$ $\|\Omega\| = 801$ $\|\Omega_{test}\| = 804$	Grid	17.07 ± 0.36	0.41 ± 0.02	0.24 ± 0.02	36.77 ± 0.99	0.39 ± 0.02	0.24 ± 0.01
	Random	17.81 ± 0.30	0.41 ± 0.02	0.21 ± 0.03	39.03 ± 0.65	0.39 ± 0.02	0.21 ± 0.02
	TPE	187.91 ± 39.92	0.42 ± 0.02	0.23 ± 0.02	447.17 ± 85.49	0.40 ± 0.02	0.24 ± 0.01
	VF-iDCA	2.40 ± 0.13	0.27 ± 0.02	0.17 ± 0.01	11.01 ± 1.26	0.19 ± 0.02	0.18 ± 0.01
	LDMMA	1.24 ± 0.12	0.20 ± 0.02	0.15 ± 0.08	8.04 ± 0.71	0.15 ± 0.05	0.17 ± 0.01
w1a $p = 300$ $\|\Omega\| = 1236$ $\|\Omega_{test}\| = 1241$	Grid	20.08 ± 0.33	0.59 ± 0.10	0.41 ± 0.14	104.47 ± 2.99	0.06 ± 0.01	0.03 ± 0.00
	Random	20.30 ± 0.18	0.55 ± 0.07	0.31 ± 0.08	147.88 ± 8.64	0.05 ± 0.00	0.02 ± 0.00
	TPE	85.80 ± 13.95	0.64 ± 0.13	0.45 ± 0.11	682.35 ± 17.52	0.06 ± 0.01	0.03 ± 0.00
	VF-iDCA	4.32 ± 0.23	0.03 ± 0.02	0.03 ± 0.00	25.37 ± 3.10	0.01 ± 0.00	0.03 ± 0.00
	LDMMA	2.19 ± 0.24	0.01 ± 0.00	0.01 ± 0.00	15.25 ± 2.90	0.01 ± 0.00	0.02 ± 0.00