AN EFFICIENT AUGMENTED LAGRANGIAN-BASED METHOD FOR LINEAR EQUALITY-CONSTRAINED LASSO

Zengde Deng[†]*, Man-Chung Yue*[‡]*, Anthony Man-Cho So*[†]

[†]Department of Systems Engineering and Engineering Management, CUHK, Hong Kong
[‡]Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong

ABSTRACT

Variable selection is one of the most important tasks in statistics and machine learning. To incorporate more prior information about the regression coefficients, various constrained Lasso models have been proposed in the literature. Compared with the classic (unconstrained) Lasso model, the algorithmic aspects of constrained Lasso models are much less explored. In this paper, we demonstrate how the recently developed semismooth Newton-based augmented Lagrangian framework can be extended to solve a linear equality-constrained Lasso model. A key technical challenge that is not present in prior works is the lack of strong convexity in our dual problem, which we overcome by adopting a regularization strategy. We show that under mild assumptions, our proposed method will converge superlinearly. Moreover, extensive numerical experiments on both synthetic and real-world data show that our method can be substantially faster than existing first-order methods while achieving a better solution accuracy.

***Index Terms*—** constrained Lasso, augmented Lagrangian, semismooth Newton, superlinear convergence

1. INTRODUCTION

With the advent of big data era, variable selection has received great attention in statistics and machine learning. There exist a host of methods to address this problem, such as Lasso [20], SCAD [8], elastic net [24] and so on. Benefiting from the simple formulation and the powerful modeling concerning the variable selection task, Lasso has been extensively applied in various instances [2, 3]. In spite of its overwhelming success, Lasso still suffers from the limited information induced by l_1 norm. To circumvent this issue, researchers proposed the constrained Lasso model [9, 12] to incorporate more prior information. Motivated by the above discussions, we propose an efficient algorithm to tackle the constrained Lasso problem

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|^2+\lambda\|x\|_1 \quad \text{s.t.} \quad Bx=d, \tag{1}$$

where $b\in\mathbb{R}^m$ is the response vector, $A\in\mathbb{R}^{m\times n}$ is the design matrix, and $B\in\mathbb{R}^{s\times n}, d\in\mathbb{R}^s$ are given constraints.

An important example which falls into the constrained Lasso problem is Lasso with sum-to-zero constraint, i.e. $e^Tx=0$. This constraint has been adopted in microbiome data regression [19] and variable selection [15] where the covariates come from compositional data. Another example widely used in statistics is the generalized Lasso problem

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|^2+\lambda\|Dx\|_1, \tag{2}$$

where $D\in\mathbb{R}^{p\times n}$. When rank$(D)=p$ and $p\le n$, Tibshirani [21] has derived that (2) can be transformed into a Lasso problem. In fact, (2) is a special case of constrained Lasso with $d=0$ [9, 12] when $p\ge n$ and D has full column rank n.

Our contributions. In this paper, we propose a semismooth Newton augmented Lagrangian method to solve the constrained Lasso problem. To fully exploit the sparsity structure, we focus on the dual formulation of our problem and propose an inexact augmented Lagrangian method. The main challenge lies in how to solve the subproblem of augmented Lagrangian method efficiently. To overcome this difficulty, we apply the semismooth Newton method to solve the inner subproblem. Our numerical experiments indicate that we only need tens of outer iterations. For the subproblem, we need about ten iterations to reach the desired accuracy. Hence, the total running time is small. The key insights behind this impressive performance are three-fold: (a) Regarding the outer loop, we have superlinear convergence to achieve highly accurate solution. (b) Besides, we attain fast convergence in the inner subproblem solver and hence the total iteration number is still small. (c) When solving the inner subproblem, we extensively exploit the underlying sparsity structure of the generalized Hessian in the subproblem (refered to as second-order sparsity [13]) to greatly reduce the computational cost. In summary, not only can we prove the theoretical effectiveness of our algorithm, but also provide highly efficient implementations

2. RELATED WORK

Recently, semismooth Newton augmented Lagrangian method is attracting more and more attention due to its fast convergence and good experimental performance. Such a method has been used to tackle semidefinite programming [23], Lasso [13] and fused Lasso [14] and so on. Our work is closely related to [13], which proposed a semismooth Newton augmented Lagrangian method to solve the (standard) Lasso. However,

Algorithm 1 Inexact ALM for (D)

1: **Input:** u^0, v^0, w^0, x^0.
2: **for** $k = 0, 1, \ldots$ **do**
3: Get an approximate solution

$$z^{k+1} := (u^{k+1}, v^{k+1}, w^{k+1}) \approx \underset{z:=(u,v,w)}{\operatorname{argmin}} \left\{\Theta_k(z) := \mathcal{L}_{\sigma_k}(u, v, w; x^k)\right\}. \tag{3}$$

4: Update x by $x^{k+1} = x^k - \sigma_k(A^T u^{k+1} - B^T v^{k+1} + w^{k+1})$ and update $\sigma_{k+1} \uparrow \sigma_\infty \leq \infty$.
5: **end for**

it is not clear whether the constrained Lasso problem can be solved by the same method, as it involves multiple nonsmooth terms including both the ℓ_1 norm and linear equality constraint set. Moreover, unlike the setting considered in [13], our dual problem is not strongly convex, which hinders fast solution of the inner subproblem. We circumvent this by using a modified semismooth Newton method and give the corresponding convergence analysis.

It should be noted that some first-order methods can also be applied to solve (1), which includes first-order primal-dual methods [4], linearized augmented Lagrangian method [22], and alternating direction method of multipliers (ADMM) [1]. Moreover, three-operator-splitting [7] and proximal-proximal gradient method [18] are also proposed to handle such kind of problems with multiple nonsmooth terms.

3. PROBLEM FORMULATION AND AN AUGMENTED LAGRANGIAN METHOD

We propose an augmented Lagrangian method to solve the dual of problem (1) in this section. We first derive the dual problem (D) of our constrained Lasso problem. Our problem (1) can be written in the following way:

$$\min_{x\in\mathbb{R}^n}\{f(x) = h(Ax) + p(x)\} \quad \text{s.t.} \quad Bx = d, \tag{P}$$

where $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{s\times n}$, $d \in \mathbb{R}^s$, $h(x) = \frac{1}{2}\|x - b\|^2$, and $p(x) = \lambda\|x\|_1$. The dual problem of (P) can be written as

$$\begin{aligned}\min_{u\in\mathbb{R}^m, v\in\mathbb{R}^s, w\in\mathbb{R}^n} &\{g(z) := h^*(u) - \langle v, d\rangle + p^*(w)\}\\ &\text{s.t. } A^T u - B^T v + w = 0,\end{aligned} \tag{D}$$

where $h^*(u) = \frac{1}{2}\|u\|^2 + b^T u$ and $p^*(w) = \mathbb{I}_{\{\|w\|_\infty\leq\lambda\}}$ are conjugate functions of h and p respectively, $\mathbb{I}$ is the indicator function, and we denote $z := (u, v, w)$.

The augmented Lagrangian function for (D) is given by

$$\mathcal{L}_\sigma(u, v, w; x) = l(u, v, w; x) + \frac{\sigma}{2}\|A^T u - B^T v + w\|^2,$$

where $l(u, v, w; x)$ is the Lagrangian function given as $l(z; x) \equiv l(u, v, w; x) := h^*(u) - \langle v, d\rangle + p^*(w) - \langle x, A^T u - B^T v + w\rangle$. We propose Algorithm 1 to solve (D). For efficiency, we solve the subproblem (3) inexactly. We use one of the following stopping criteria by Rockafellar [16]:

$$\Theta_k(z^{k+1}) - \inf\Theta_k \leq \varepsilon_k^2/2\sigma_k, \tag{A}$$

$$\Theta_k(z^{k+1}) - \inf\Theta_k \leq \zeta_k^2\|x^{k+1} - x^k\|^2/2\sigma_k, \tag{B}$$

where $\sum_{k=0}^\infty \varepsilon_k < \infty$ and $\sum_{k=0}^\infty \zeta_k < \infty$.

4. INEXACT SEMISMOOTH NEWTON METHOD TO SOLVE ALM SUBPROBLEM

The main challenge of solving (D) via the inexact ALM lies in how to solve the subproblem (3) efficiently. Due to the nonsmoothness of the subproblem, we propose a semismooth Newton method to solve ALM subproblem (3) and provide a highly efficient implementation by exploiting structures.

We first define the proximal mapping associated with p as $\text{Prox}_p(x) := \operatorname{argmin}_u \left\{p(u) + \frac{1}{2}\|u - x\|^2\right\}$. For a fixed x and given σ, we consider

$$\min_{y,w}\Theta(y, w) := \mathcal{L}_\sigma(y, w; x), \tag{4}$$

where $y = (u, v) \in \mathbb{R}^{m+s}$, and for convenience we set $\bar{h}^*(y) = h^*(u) - \langle v, d\rangle$. Then, we define $\theta(y)$ by

$$\begin{aligned}\theta(y) = \inf_w \mathcal{L}_\sigma(y, w; x) &= \bar{h}^*(y) + p^*\left(\text{Prox}_{p^*/\sigma}(x/\sigma - \bar{A}^T y)\right)\\ &+ \frac{1}{2\sigma}\left\|\text{Prox}_{\sigma p}(x - \sigma(\bar{A}^T y)\right\|^2 - \frac{1}{2\sigma}\|x\|^2\\ &= \bar{h}^*(y) + \sigma E_{p^*/\sigma}(x/\sigma - \bar{A}^T y) - \frac{1}{2\sigma}\|x\|^2,\end{aligned}$$

where the last equality follows from the Moreau decomposition, $E_{p^*/\sigma}$ is the Moreau envelope of p^*/σ, and $\bar{A} = [A^T, -B^T]^T$. Hence, if we let $(\tilde{y}, \tilde{w}) = \operatorname{argmin}\Theta(y, w)$, then $(\tilde{y}, \tilde{w})$ can be computed in the following manner:

$$\begin{cases}\tilde{y} = \operatorname{argmin}\theta(y),\\ \tilde{w} = \text{Prox}_{p^*/\sigma}(x/\sigma - \bar{A}^T\tilde{y}).\end{cases} \tag{5}$$

Since the Moreau envelope $E_{p^*/\sigma}$ is continuously differentiable [17], θ is a convex continuously differentiable function in y with

$$\nabla\theta(y) = \begin{bmatrix}\nabla h^*(u) - A\text{Prox}_{\sigma p}(x - \sigma(\bar{A}^T y))\\ -d + B\text{Prox}_{\sigma p}(x - \sigma(\bar{A}^T y))\end{bmatrix}.$$

Moreover, (5) is equivalent to the following:

$$\nabla\theta(y) = 0. \tag{6}$$

For any $y \in \text{dom}(y)$, we define

$$\hat{\partial}^2\theta(y) := H + \sigma\bar{A}\partial\text{Prox}_{\sigma p}(x - \sigma(\bar{A}^T y))\bar{A}^T,$$

where $H = \begin{bmatrix}\nabla^2 h^*(u) & \\ & \mathbf{0}\end{bmatrix}$ and $\partial\text{Prox}_{\sigma p}(x - \sigma(\bar{A}^T y))$ is the Clarke subdifferential [6] of $\text{Prox}_{\sigma p}(\cdot)$ at $x - \sigma(\bar{A}^T y)$. From [10], we know that

$$\hat{\partial}^2\theta(y)(d_u, d_v) = \partial^2\theta(y)(d_u, d_v),$$

where $\partial^2\theta(\cdot)$ denotes the generalized Hessian of $\theta(\cdot)$. Define

$$V := H + \sigma\bar{A}Q\bar{A}^T, \tag{7}$$

with $Q \in \partial\text{Prox}_{\sigma p}(x - \sigma(\bar{A}^T y))$, then we have $V \in \hat{\partial}^2\theta(y)$.

Note that in our problem, $h^*(\cdot)$ is twice continuous differentiable and $\text{Prox}_{\lambda\|x\|_1}(\cdot)$ is piecewise linear, which are all strongly semismooth [13]. Hence we give a semismooth Newton (SSN) method in Algorithm 2 to solve equation (6) .

Algorithm 2 Semismooth Newton (SSN) for subproblem

1: **Input:** Given $\mu \in (0, 0.5)$, $\bar{\eta} \in (0, 1)$, $\tau \in (0, 1]$, $\tau_1, \tau_2 \in (0, 1)$ and $\delta \in (0, 1)$. Choose $y^0 = (u^0, v^0)$.
2: **for** $j = 0, 1, \ldots$ **do**
3: Choose $Q^j \in \partial \text{Prox}_{\sigma p}(x - \sigma(\bar{A}^T y^j))$. Let V_j be given as in (7) and $\epsilon_j = \tau_1 \min\{\tau_2, \|\nabla\theta(y^j)\|\}$. Solve the following linear system

$$V_j d_y^j + \epsilon_j (0, d_v^j) = -\nabla\theta(y^j), \tag{8}$$

exactly where $d_y^j = (d_u^j, d_v^j)$ or by CG algorithm to find an approximate solution such that

$$\|V_j d_y^j + \epsilon_j (0, d_v^j) + \nabla\theta(y^j)\| \le \min(\bar{\eta}, \|\nabla\theta(y^j\|^{1+\tau}).$$

4: (Line search) Set $\alpha_j = \delta^{l_j}$, where l_j is the first nonnegative integer l for which

$$\theta(y^j + \delta^l d_y^j) \le \theta(u^j, v^j) + \mu\delta^l \langle \nabla\theta(y^j), d_y^j \rangle.$$

5: Set $u^{j+1} = u^j + \alpha_j d_u^j$ and $v^{j+1} = v^j + \alpha_j d_v^j$.
6: **end for**

Due to lack of strongly convexity, which is a setting different from [13], we introduce a regularized term on v to obtain (8).

4.1. Efficient implementation of SSN

As mentioned before, the key step of the whole algorithmic framework is how to solve (3) quickly. We use the semismooth Newton method to tackle (3) and the main computational cost lies in (8), which computes the inexact Newton direction. Thus we will give efficient implementations to compute (8).

Recall the definition of V in (7) and $h^*(u) = \frac{1}{2}\|u\|^2 + b^T u$ in our problem, we have $H = \text{diag}(\mathbf{1}_m, \mathbf{0}_s)$, where $\mathbf{1}_m$ denotes the all-ones m-dimensional vector and the same for $\mathbf{0}_s$, $\text{diag}(x)$ is the diagonal matrix with vector x. Hence, we rewrite (8) as

$$(H_\epsilon + \sigma \bar{A} Q \bar{A}^T) d_y = -\nabla\theta(y), \tag{9}$$

where $H_\epsilon := H + \text{diag}(\mathbf{0}_m, \epsilon\mathbf{1}_s) = \text{diag}(\mathbf{1}_m, \epsilon\mathbf{1}_s)$. Since H_ϵ is positive definite, our linear system is well defined.

Before solving (9), we do Cholesky decomposition on H_ϵ via $H_\epsilon = LL^T$ with $L = L^T = \text{diag}(\mathbf{1}_m, \sqrt{\epsilon}\mathbf{1}_s)$. By some basic calculations, we rewrite (9) as

$$(\mathbf{I}_{m+s} + \sigma \hat{A} Q \hat{A}^T) \hat{d}_y = -\nabla\hat{\theta}(y), \tag{10}$$

where $\hat{A} = L^{-1}\bar{A}$, $\hat{d}_y = L^{-1} d_y$, $\nabla\hat{\theta}(y) = L^{-1}\nabla\theta(y)$ and $\mathbf{I}_{m+s}$ denotes the identity matrix. Note that the cost of computing both $\bar{A}Q\bar{A}$ and $\hat{A}Q\hat{A}^T$ are $\mathcal{O}((m+s)^2 n)$. Therefore, the matrix multiplication can be computational prohibitive when n is large. Fortunately, we can overcome this difficulty by exploiting the sparsity structure of Q which we call this *second-order sparsity* of our problem in the following way.

For the subdifferential of proximal mapping, we can always choose $Q \in \partial \text{Prox}_{\sigma\lambda\|x\|_1}(x)$ to be $Q = \text{diag}(q)$, a diagonal matrix whose i-th element is given by

$$q_i = \begin{cases} 1, & \text{if} \quad |x_i| > \sigma\lambda, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Set $\mathcal{J} = \{j : |x_j| > \sigma\lambda\}$ with cardinality $|\mathcal{J}| = r$. By utilizing the diagonal structure of Q, we can write

$$\bar{A} Q \bar{A}^T = \bar{A}_{\mathcal{J}} \bar{A}_{\mathcal{J}}^T, \quad \hat{A} Q \hat{A}^T = \hat{A}_{\mathcal{J}} \hat{A}_{\mathcal{J}}^T, \tag{12}$$

where $\bar{A}_{\mathcal{J}} \in \mathbb{R}^{(m+s)\times r}$ is the submatrix of $\bar{A}$ with those columns contained in $\mathcal{J}$ preserved and the same for $\hat{A}_{\mathcal{J}}$.

Now we analyze the reduction of computational cost by exploring the second-order sparsity of the problem. By utilizing (12), we can reduce the cost of computing $\bar{A}Q\bar{A}$ and $\hat{A}Q\hat{A}^T$ from $\mathcal{O}((m+s)^2 n)$ to $\mathcal{O}((m+s)^2 r)$. Due to the sparsity-inducing property of $p(x) = \lambda\|x\|_1$, r is usually much smaller than n, we greatly reduce the computational cost. Consequently, the total computational cost of solving (9) reduces from $\mathcal{O}((m+s)^2(m+s+n))$ to $\mathcal{O}((m+s)^2(m+s+r))$, meaning that the computational cost has no relationship with n. Thus, even for large dimension n, we can tackle the linear system (9) by Cholesky factorization.

In fact, when $r \ll m+s$, we can also directly invert the matrix using the Sherman-Morrison-Woodbury formula:

$$(\mathbf{I}_{m+s} + \sigma \hat{A} Q \hat{A}^T)^{-1} = (\mathbf{I}_{m+s} + \sigma \hat{A}_{\mathcal{J}} \hat{A}_{\mathcal{J}}^T)^{-1}$$
$$= \mathbf{I}_{m+s} - \hat{A}_{\mathcal{J}} (\sigma^{-1}\mathbf{I}_r + \hat{A}_{\mathcal{J}}^T \hat{A}_{\mathcal{J}})^{-1} \hat{A}_{\mathcal{J}}^T.$$

As a result, the total computational cost to solve (9) can be further reduced from $\mathcal{O}((m+s)^2(m+s+r))$ to $\mathcal{O}(r^2(m+s+r))$. Note that whichever way we choose to solve (9), the computational cost only depends on $m+s$. Thus, when $m+s$ is not too large (smaller than 10^4), we can always solve (9) exactly by Cholesky factorization or by computing the inverse. Otherwise, we can choose CG to solve (9) inexactly.

5. CONVERGENCE ANALYSIS

We first provide the convergence result and corresponding superlinear convergence rate of Algorithm 1 in the following theorems. More details and proofs about these two theorems are deferred to the full version of our paper.

Theorem 1. *Suppose that the solution set X_P^* to (P) is nonempty. Let $\{(z^k, x^k)\}$ be the sequence generated by Algorithm 1 with stopping criterion (A). Then the sequence $\{x^k\}$ is bounded and converges to some point $x^\infty \in X_P^*$.*

Theorem 2. *Suppose that both (P) and (D) have optimal solution sets X_P^* and Z_D^* respectively. Let $\{(z^k, x^k)\}$ be the sequence generated by Algorithm 1 with stopping criterion (B). Then the sequence $\{x^k\}$ converges to some $x^\infty \in X_P^*$ superlinearly.*

Moreover, we establish the convergence of Algorithm 2 under a mild assumption that $\text{rank}(B) = s$. The proof closely follows that of Theorem 3.4 of [23].

Theorem 3. *Suppose that rank$(B) = s$. Then Algorithm 2 is well defined and any accumulation point $(\hat{u}, \hat{v})$ is an optimal solution to problem (6).*

6. NUMERICAL EXPERIMENTS

In this section, we evaluate the performance of our algorithm for solving (1) on both synthetic and real datasets. We com-

Table 1: Performance of our SSNAL method (a), primal dual method (b), linearized ALM (c), ADMM (d) and PPG (e) with generalized Lasso problem on synthetic and real datasets. 'nnz' denotes the number of nonzero elements in the optimal solution obtained by our baseline method.

(a) synthetic datasets

	λ_l	nnz	running time (seconds)				
size $m; n$			a	b	c	d	e
500;5000	10^{-2}	499	**1.0**	19	29	21	19
	10^{-3}	505	**1.5**	19	30	21	19
	10^{-4}	508	**2.0**	20	29	21	19
800;8000	10^{-2}	777	**2.7**	49	65	53	48
	10^{-3}	784	**3.8**	49	65	52	48
	10^{-4}	789	**5.6**	49	65	51	49
1000;10000	10^{-2}	970	**4.8**	75	95	80	73
	10^{-3}	978	**6.1**	74	94	81	70
	10^{-4}	982	**7.7**	70	89	74	69

(b) real datasets

	λ_l	nnz	running time (seconds)				
problem name $m; n$			a	b	c	d	e
abalone7	10^{-3}	46	**49**	176	191	258	177
4177;6435	10^{-4}	78	**12**	178	190	102	175
bodyfat5	10^{-3}	44	**1.5**	23	45	17	22
252;11628	10^{-4}	63	**1.2**	24	47	3.3	22
housing5	10^{-3}	103	**1.9**	32	49	31	32
506;8568	10^{-4}	223	**2.0**	34	51	12	31
space_ga9	10^{-3}	46	**14**	106	115	150	105
3107;5005	10^{-4}	73	**6.4**	105	114	32	104

pare with four state-of-the-art first-order methods: primal-dual method [4], linearized ALM [22], ADMM [1], proximal-proximal gradient (PPG) [18].

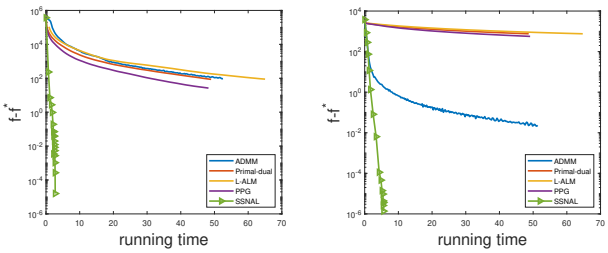
We set the penalty parameter λ in the constrained Lasso problem as $\lambda = \lambda_l \|A^T b\|_\infty$, where $0 < \lambda_l < 1$. The accuracy of solution $\{x, u, v, w\}$ generated by our algorithm is measured by $\eta_{\text{cLasso}} = \max\{\eta_P, \eta_D\}$, where $\eta_P = \frac{\|Bx-d\|}{1+\|d\|}$ and $\eta_D = \|A^T u - B^T v + w\|$ are the primal and dual feasibility. We stop our algorithm when $\eta_{\text{cLasso}} < \varepsilon$ for a given tolerance ε and stop other compared algorithms when both the primal and dual residuals are smaller than ε. Before the comparison, we run our algorithm with high accuracy $\varepsilon = 10^{-10}$ and set this optimal value as the baseline. The optimality gap is measured by $\eta_{\text{gap}} = f(x) - f(x^*)$. For our numerical experiments, we set $\varepsilon = 10^{-6}$. All the algorithms will be stopped when they reach the maximum iteration number, which is set at 100 for our algorithm and at 10000 for other algorithms. The codes were written in MATLAB and run on a PC with i5-6500 CPU at 3.2 GHz with 16 GB memory.

For both synthetic and real datasets, three scenarios are tested: (a) sum to zero constraint; (b) B and d are randomly generated; (c) generalized Lasso problem. Due to limited space, we present last one here, and full details of the experiments for all three scenarios are deferred to the full version.

Generalized Lasso. We transform the generalized Lasso problem (2) to an equivalent constrained Lasso problem using techniques from [9] and construct $D = [D_1^T, D_2^T]^T$, where $D_1 = \mathbf{I}_n$ and D_2 is an $s \times n$ random matrix and we omit the details here.

Synthetic data. In this subsection we display the performance of our algorithm on synthetic datasets. We generate $A \in \mathbb{R}^{m\times n}$ from independent and identical (iid) standard normal distribution and $b = A\mathring{x} + \varrho$, where $\varrho \in N(0, 0.001 * \mathbf{I}_m)$ and $\mathring{x}$ is a sparse vector. We set $n = 10m$ with $m = 200, 300, 500, 800, 1000$ and $\lambda_l = 10^{-2}, 10^{-3}, 10^{-4}$.

In Table 1a we summarize part of the numerical results. Observe that our algorithm is 5-10 times faster than other algorithms for all choices of λ_l. We plot the optimality gap with running time in Figure 1 for the case $m = 800, n = 8000$ and $s = 30$. The figures demonstrate that our algorithm is superior to other methods both in terms of solution accuracy and running time.



(a) $\lambda_l = 10^{-2}$ (b) $\lambda_l = 10^{-4}$

Fig. 1: Generalized Lasso problem on synthetic dataset with $m = 800, n = 8000, s = 30$ and $\lambda_l = 10^{-2}, 10^{-4}$.

Real data. In this subsection we test on LIBSVM datasets [5] with $\lambda_l = 10^{-3}$ and 10^{-4}. We preprocess the datasets to expand the original features based on polynomial basis functions as stated in [11]. For example, **abalone7** means that we expand the feature of **abalone** by an order 7 polynomial basis function. We present part of numerical results in Table 1b. It is worth to note that our algorithm can also be 5-10 times faster than other first-order methods on real datasets.

7. CONCLUSION

In this paper, we propose a semismooth Newton augmented Lagrangian method to solve the linear-equality constrained Lasso problem and establish convergence results for both outer loop and inner subproblem solver. By exploiting the sparsity structure of the problem, we provide efficient implementations to solve the subproblem and greatly reduce the computational cost. Extensive numerical experiments demonstrate both the efficiency and accuracy of our algorithm. As a future work, we plan to extend our algorithmic framework to handle more general constraints such as inequality constraints.

Acknowledgement. A. M.-C. So is partially supported by the Hong Kong Research Grants Council (RGC) General Research Fund (GRF) Project CUHK 14203218 and the CUHK Research Sustainability of Major RGC Funding Schemes Project 3133236.

8. REFERENCES

[1] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[2] Kenneth P Burnham and David R Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer Science & Business Media, 2003.

[3] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

[4] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[5] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[6] Frank H Clarke. *Optimization and Nonsmooth Analysis*, volume 5. SIAM, 1990.

[7] Damek Davis and Wotao Yin. A three-operator splitting scheme and its optimization applications. *Set-Valued and Variational Analysis*, 25(4):829–858, 2017.

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

[9] Brian R Gaines, Juhyun Kim, and Hua Zhou. Algorithms for fitting the constrained Lasso. *Journal of Computational and Graphical Statistics*, 27(4):861–871, 2018.

[10] Jean-Baptiste Hiriart-Urruty, Jean-Jacques Strodiot, and V Hien Nguyen. Generalized Hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data. *Applied Mathematics and Optimization*, 11(1):43–56, 1984.

[11] Ling Huang, Jinzhu Jia, Bin Yu, Byung-Gon Chun, Petros Maniatis, and Mayur Naik. Predicting execution time of computer programs using sparse polynomial regression. In *Advances in Neural Information Processing Systems*, pages 883–891, 2010.

[12] Gareth M James, Courtney Paulson, and Paat Rusmevichientong. Penalized and constrained regression. *Unpublished Manuscript, available at http://www-bcf. usc. edu/˜ gareth/research/Research. html*, 2013.

[13] Xudong Li, Defeng Sun, and Kim-Chuan Toh. A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems. *SIAM Journal on Optimization*, 28(1):433–458, 2018.

[14] Xudong Li, Defeng Sun, and Kim-Chuan Toh. On efficiently solving the subproblems of a level-set method for fused Lasso problems. *SIAM Journal on Optimization*, 28(2):1842–1866, 2018.

[15] Wei Lin, Pixu Shi, Rui Feng, and Hongzhe Li. Variable selection in regression with compositional covariates. *Biometrika*, 101(4):785–797, 2014.

[16] R Tyrrell Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1(2):97–116, 1976.

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

[18] Ernest K Ryu and Wotao Yin. Proximal-proximal-gradient method. *arXiv preprint arXiv:1708.06908*, 2017.

[19] Pixu Shi, Anru Zhang, and Hongzhe Li. Regression analysis for microbiome compositional data. *Annals of Applied Statistics*, 10(2):1019–1040, 2016.

[20] Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[21] Ryan Joseph Tibshirani. *The solution path of the generalized lasso*. Stanford University, 2011.

[22] Junfeng Yang and Xiaoming Yuan. Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of Computation*, 82(281):301–329, 2013.

[23] Xinyuan Zhao, Defeng Sun, and Kim-Chuan Toh. A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM Journal on Optimization*, 20(4):1737–1765, 2010.

[24] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.