QUIC & DIRTY: A Quadratic Approximation Approach for Dirty Statistical Models

Cho-Jui Hsieh, Inderjit S. Dhillon, Pradeep Ravikumar
University of Texas at Austin
Austin, TX 78712 USA
{cjhsieh,inderjit,pradeepr}@cs.utexas.edu

Stephen Becker
University of Colorado at Boulder
Boulder, CO 80309 USA
stephen.becker@colorado.edu

Peder A. Olsen
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598 USA
pederao@us.ibm.com

Abstract

In this paper, we develop a family of algorithms for optimizing "superposition-structured" or "dirty" statistical estimators for high-dimensional problems involving the minimization of the sum of a smooth loss function with a hybrid regularization. Most of the current approaches are first-order methods, including proximal gradient or Alternating Direction Method of Multipliers (ADMM). We propose a new family of second-order methods where we approximate the loss function using quadratic approximation. The superposition structured regularizer then leads to a subproblem that can be efficiently solved by alternating minimization. We propose a general active subspace selection approach to speed up the solver by utilizing the low-dimensional structure given by the regularizers, and provide convergence guarantees for our algorithm. Empirically, we show that our approach is more than 10 times faster than state-of-the-art first-order approaches for the latent variable graphical model selection problems and multi-task learning problems when there is more than one regularizer. For these problems, our approach appears to be the first algorithm that can extend active subspace ideas to multiple regularizers.

1 Introduction

From the considerable amount of recent research on high-dimensional statistical estimation, it has now become well understood that it is vital to impose structural constraints upon the statistical model parameters for their statistically consistent estimation. These structural constraints take the form of sparsity, group-sparsity, and low-rank structure, among others; see [18] for unified statistical views of such structural constraints. In recent years, such "clean" structural constraints are frequently proving insufficient, and accordingly there has been a line of work on "superposition-structured" or "dirty model" constraints, where the model parameter is expressed as the sum of a number of parameter components, each of which have their own structure. For instance, [4, 6] consider the estimation of a matrix that is neither low-rank nor sparse, but which can be decomposed into the sum of a low-rank matrix and a sparse outlier matrix (this corresponds to robust PCA when the matrix-structured parameter corresponds to a covariance matrix). [5] use such matrix decomposition to estimate the structure of latent-variable Gaussian graphical models. [15] in turn use a superposition of sparse and group-sparse structure for multi-task learning. For other recent work on such superposition-structured models, see [1, 7, 14]. For a unified statistical view of such superposition-structured models, and the resulting classes of M-estimators, please see [27].

Consider a general superposition-structured parameter $\bar{\boldsymbol{\theta}} := \sum_{r=1}^{k} \boldsymbol{\theta}^{(r)}$, where $\{\boldsymbol{\theta}^{(r)}\}_{r=1}^{k}$ are the parameter-components, each with their own structure. Let $\{\mathcal{R}^{(r)}(\cdot)\}_{r=1}^{k}$ be regularization functions suited to the respective parameter components, and let $\mathcal{L}(\cdot)$ be a (typically non-linear) loss function

that measures the goodness of fit of the superposition-structured parameter $\bar{\boldsymbol{\theta}}$ to the data. We now have the notation to consider a popular class of M-estimators studied in the papers above for these superposition-structured models:

$$\min_{\{\boldsymbol{\theta}^{(r)}\}_{r=1}^{k}} \left\{ \mathcal{L}\Big(\sum_{r} \boldsymbol{\theta}^{(r)} \Big) + \sum_{r} \lambda_r \mathcal{R}^{(r)}(\boldsymbol{\theta}^{(r)}) \right\} := F(\boldsymbol{\theta}), \tag{1}$$

where $\{\lambda_r\}_{r=1}^{k}$ are regularization penalties. In (1), the overall regularization contribution is separable in the individual parameter components, but the loss function term itself is not, and depends on the sum $\bar{\boldsymbol{\theta}} := \sum_{r=1}^{k} \boldsymbol{\theta}^{(r)}$. Throughout the paper, we use $\bar{\boldsymbol{\theta}}$ to denote the overall superposition-structured parameter, and $\boldsymbol{\theta} = [\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(k)}]$ to denote the concatenation of all the parameters.

Due to the wide applicability of this class of M-estimators in (1), there has been a line of work on developing efficient optimization methods for solving special instances of this class of M-estimators [14, 26], in addition to the papers listed above. In particular, due to the superposition-structure in (1) and the high-dimensionality of the problem, this class seems naturally amenable to a proximal gradient descent approach or the ADMM method [2, 17]; note that these are first-order methods and are thus very scalable.

In this paper, we consider instead a proximal Newton framework to minimize the M-estimation objective in (1). Specifically, we use iterative quadratic approximations, and for each of the quadratic subproblems, we use an alternating minimization approach to individually update each of the parameter components comprising the superposition-structure. Note that the Hessian of the loss might be structured, as for instance with the logdet loss for inverse covariance estimation and the logistic loss, which allows us to develop very efficient second-order methods. Even given this structure, solving the regularized quadratic problem in order to obtain the proximal Newton direction is too expensive due to the high dimensional setting. The key **algorithmic contribution** of this paper is in developing a general active subspace selection framework for general decomposable norms, which allows us to solve the proximal Newton steps over a significantly reduced search space. We are able to do so by leveraging the structural properties of decomposable regularization functions in the M-estimator in (1).

Our other key contribution is **theoretical**. While recent works [16, 21] have analyzed the convergence of proximal Newton methods, the superposition-structure here poses a key caveat: since the loss function term only depends on the sum of the individual parameter components, the Hessian is not positive-definite, as is required in previous analyses of proximal Newton methods. The theoretical analysis [9] relaxes this assumption by instead assuming the loss is self-concordant but again allows at most one regularizer. Another key theoretical difficulty is our use of active subspace selection, where we do not solve for the vanilla proximal Newton direction, but solve the proximal Newton step subproblem only over a *restricted subspace*, which moreover varies with each step. We deal with these issues and show **super-linear convergence** of the algorithm when the sub-problems are solved exactly. We apply our algorithm to two real world applications: latent Gaussian Markov random field (GMRF) structure learning (with low-rank + sparse structure), and multitask learning (with sparse + group sparse structure), and demonstrate that our algorithm is more than **ten** times faster than state-of-the-art methods.

Overall, our algorithmic and theoretical developments open up the state of the art but forbidding class of M-estimators in (1) to very large-scale problems.

Outline of the paper. We begin by introducing some background in Section 2. In Section 3, we propose our quadratic approximation framework with active subspace selection for general dirty statistical models. We derive the convergence guarantees of our algorithm in Section 4. Finally, in Section 5, we apply our model to solve two real applications, and show experimental comparisons with other state-of-the-art methods.

2 Background and Applications

Decomposable norms. We consider the case where all the regularizers $\{\mathcal{R}^{(r)}\}_{r=1}^{k}$ are decomposable norms $\|\cdot\|_{\mathcal{A}_r}$. A norm $\|\cdot\|$ is decomposable at $\boldsymbol{x}$ if there is a subspace $\mathcal{T}$ and a vector $\boldsymbol{e} \in \mathcal{T}$ such that the sub differential at $\boldsymbol{x}$ has the following form:

$$\partial \|\boldsymbol{x}\|_r = \{\boldsymbol{\rho} \in \mathbb{R}^n \mid \Pi_{\mathcal{T}}(\boldsymbol{\rho}) = \boldsymbol{e} \text{ and } \|\Pi_{\mathcal{T}^\perp}(\boldsymbol{\rho})\|_{\mathcal{A}_r}^* \leq 1\}, \tag{2}$$

where $\Pi_{\mathcal{T}}(\cdot)$ is the orthogonal projection onto $\mathcal{T}$, and $\|\boldsymbol{x}\|^* := \sup_{\|\boldsymbol{a}\| \leq 1} \langle \boldsymbol{x}, \boldsymbol{a} \rangle$ is the dual norm of $\|\cdot\|$. The decomposable norm was defined in [3, 18], and many interesting regularizers belong to this category, including:

• Sparse vectors: for the ℓ_1 regularizer, $\mathcal{T}$ is the span of all points with the same support as $\boldsymbol{x}$.

• Group sparse vectors: suppose that the index set can be partitioned into a set of $N_{\mathcal{G}}$ disjoint groups, say $\mathcal{G} = \{G_1, \ldots, G_{N_{\mathcal{G}}}\}$, and define the (1,$\alpha$)-group norm by $\|\boldsymbol{x}\|_{1,\alpha} := \sum_{t=1}^{N_{\mathcal{G}}} \|\boldsymbol{x}_{G_t}\|_\alpha$. If S_G denotes the subset of groups where $\boldsymbol{x}_{G_t} \neq 0$, then the subgradient has the following form:

$$\partial\|\boldsymbol{x}\|_{1,\alpha} := \{\boldsymbol{\rho} \mid \boldsymbol{\rho} = \sum_{t \in S_G} \boldsymbol{x}_{G_t}/\|\boldsymbol{x}_{G_t}\|_\alpha^* + \sum_{t \notin S_G} \boldsymbol{m}_t\},$$

where $\|\boldsymbol{m}_t\|_\alpha^* \leq 1$ for all $t \notin S_G$. Therefore, the group sparse norm is also decomposable with

$$\mathcal{T} := \{\boldsymbol{x} \mid \boldsymbol{x}_{G_t} = 0 \text{ for all } t \notin S_G\}. \tag{3}$$

• Low-rank matrices: for the nuclear norm regularizer $\|\cdot\|_*$, which is defined to be the sum of singular values, the subgradient can be written as

$$\partial\|X\|_* = \{UV^T + W \mid U^T W = 0, WV = 0, \|W\|_2 \leq 1\},$$

where $\|\cdot\|_2$ is the matrix 2 norm and U, V are the left/right singular vectors of X corresponding to *non-zero* singular values. The above subgradient can also be written in the decomposable form (2), where $\mathcal{T}$ is defined to be span($\{\boldsymbol{u}_i \boldsymbol{v}_j^T\}_{i,j=1}^k$) where $\{\boldsymbol{u}_i\}_{i=1}^k, \{\boldsymbol{v}_i\}_{i=1}^k$ are the columns of U and V.

Applications. Next we discuss some widely used applications of superposition-structured models, and the corresponding instances of the class of M-estimators in (1).

• Gaussian graphical model with latent variables: let Θ denote the precision matrix with corresponding covariance matrix $\Sigma = \Theta^{-1}$. [5] showed that the precision matrix will have a low rank + sparse structure when some random variables are hidden, thus $\Theta = S - L$ can be estimated by solving the following regularized MLE problem:

$$\min_{S,L: L \succeq 0, S-L \succ 0} -\log\det(S-L) + \langle S - L, \Sigma\rangle + \lambda_S\|S\|_1 + \lambda_L \operatorname{trace}(L). \tag{4}$$

While proximal Newton methods have recently become a dominant technique for solving the ℓ_1-regularized log-determinant problems [12, 10, 13, 19], our development is the first to apply proximal Newton methods to solve log-determinant problems with sparse *and* low rank regularizers.

• Multi-task learning: given k tasks, each with sample matrix $X^{(r)} \in \mathbb{R}^{n_r \times d}$ (n_r samples in the r-th task) and labels $y^{(r)}$, [15] proposes minimizing the following objective:

$$\sum_{r=1}^{k} \ell(y^{(r)}, X^{(r)}(S^{(r)} + B^{(r)})) + \lambda_S\|S\|_1 + \lambda_B\|B\|_{1,\infty}, \tag{5}$$

where $\ell(\cdot)$ is the loss function and $S^{(r)}$ is the r-th column of S.

• Noisy PCA: to recover a covariance matrix corrupted with sparse noise, a widely used technique is to solve the matrix decomposition problem [6]. In contrast to the squared loss above, an exponential PCA problem [8] would use a Bregman divergence for the loss function.

3 Our proposed framework

To perform a Newton-like step, we iteratively form quadratic approximations of the smooth loss function. Generally the quadratic subproblem will have a large number of variables and will be hard to solve. Therefore we propose a general active subspace selection technique to reduce the problem size by exploiting the structure of the regularizers $\mathcal{R}_1, \ldots, \mathcal{R}_k$.

3.1 Quadratic Approximation

Given k sets of variables $\boldsymbol{\theta} = [\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(k)}]$, and each $\boldsymbol{\theta}^{(r)} \in \mathbb{R}^n$, let $\boldsymbol{\Delta}^{(r)}$ denote perturbation of $\boldsymbol{\theta}^{(r)}$, and $\boldsymbol{\Delta} = [\boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(k)}]$. We define $g(\boldsymbol{\theta}) := \mathcal{L}(\sum_{r=1}^k \boldsymbol{\theta}^{(r)}) = \mathcal{L}(\bar{\boldsymbol{\theta}})$ to be the loss function, and $h(\boldsymbol{\theta}) := \sum_{r=1}^k \mathcal{R}^{(r)}(\boldsymbol{\theta}^{(r)})$ to be the regularization. Given the current estimate $\boldsymbol{\theta}$, we form the quadratic approximation of the smooth loss function:

$$\bar{g}(\boldsymbol{\theta} + \boldsymbol{\Delta}) = g(\boldsymbol{\theta}) + \sum_{r=1}^{k} \langle \boldsymbol{\Delta}^{(r)}, G\rangle + \frac{1}{2}\boldsymbol{\Delta}^T \mathcal{H} \boldsymbol{\Delta}, \tag{6}$$

where $G = \nabla\mathcal{L}(\bar{\boldsymbol{\theta}})$ is the gradient of $\mathcal{L}$ and $\mathcal{H}$ is the Hessian matrix of $g(\boldsymbol{\theta})$. Note that $\nabla_{\bar{\boldsymbol{\theta}}}\mathcal{L}(\bar{\boldsymbol{\theta}}) = \nabla_{\boldsymbol{\theta}^{(r)}}\mathcal{L}(\bar{\boldsymbol{\theta}})$ for all r so we simply write ∇ and refer to the gradient at $\bar{\boldsymbol{\theta}}$ as G (and similarly for ∇^2). By the chain rule, we can show that

Lemma 1. *The Hessian matrix of $g(\boldsymbol{\theta})$ is*

$$\mathcal{H} := \nabla^2 g(\boldsymbol{\theta}) = \begin{bmatrix} H & \cdots & H \\ \vdots & \ddots & \vdots \\ H & \cdots & H \end{bmatrix}, \quad H := \nabla^2 \mathcal{L}(\bar{\boldsymbol{\theta}}). \tag{7}$$

In this paper we focus on the case where H is positive definite. When it is not, we add a small constant ϵ to the diagonal of H to ensure that each block is positive definite.

Note that the full Hessian, $\mathcal{H}$, will in general, *not* be positive definite (in fact $\operatorname{rank}(\mathcal{H}) = \operatorname{rank}(H)$). However, based on its special structure, we can still give convergence guarantees (along with rate of convergence) for our algorithm. The Newton direction $\boldsymbol{d}$ is defined to be:

$$[\boldsymbol{d}^{(1)}, \ldots, \boldsymbol{d}^{(k)}] = \operatorname*{argmin}_{\boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(k)}} \bar{g}(\boldsymbol{\theta} + \boldsymbol{\Delta}) + \sum_{r=1}^{k} \lambda_r \|\boldsymbol{\theta}^{(r)} + \boldsymbol{\Delta}^{(r)}\|_{\mathcal{A}_r} := Q_{\mathcal{H}}(\boldsymbol{\Delta}; \boldsymbol{\theta}). \tag{8}$$

The quadratic subproblem (8) cannot be directly separated into k parts because the Hessian matrix (7) is not a block-diagonal matrix. Also, each set of parameters has its own regularizer, so it is hard to solve them all together. Therefore, to solve (8), we propose a block coordinate descent method. At each iteration, we pick a variable set $\boldsymbol{\Delta}^{(r)}$ where $r \in \{1, 2, \ldots, k\}$ by a cyclic (or random) order, and update the parameter set $\boldsymbol{\Delta}^{(r)}$ while keeping other parameters fixed. Assume $\boldsymbol{\Delta}$ is the current solution (for all the variable sets), then the subproblem with respect to $\boldsymbol{\Delta}^{(r)}$ can be written as

$$\boldsymbol{\Delta}^{(r)} \leftarrow \operatorname*{argmin}_{\boldsymbol{d} \in \mathbb{R}^n} \frac{1}{2} \boldsymbol{d}^T H \boldsymbol{d} + \langle \boldsymbol{d}, G + \sum_{t: r \neq t} H \boldsymbol{\Delta}^{(t)} \rangle + \lambda_r \|\boldsymbol{\theta}^{(r)} + \boldsymbol{d}\|_{\mathcal{A}_r}. \tag{9}$$

The subproblem (9) is just a typical quadratic problem with a specific regularizer, so there already exist efficient algorithms for solving it for different choices of $\|\cdot\|_{\mathcal{A}}$. For the ℓ_1 norm regularizer, coordinate descent methods can be applied to solve (9) efficiently as used in [12, 21]; (accelerated) proximal gradient descent or projected Newton's method can also be used, as shown in [19]. For a general atomic norm where there might be infinitely many atoms (coordinates), a greedy coordinate descent approach can be applied, as shown in [22].

To iterate between different groups of parameters, we have to maintain the term $\sum_{r=1}^{k} H\Delta^{(r)}$ during the Newton iteration. Directly computing $H\Delta^{(r)}$ requires $O(n^2)$ flops; however, the Hessian matrix often has a special structure so that $H\Delta^{(r)}$ can be computed efficiently. For example, in the inverse covariance estimation problem $H = \Theta^{-1} \otimes \Theta^{-1}$ where Θ^{-1} is the current estimate of covariance, and in the empirical risk minimization problem $H = XDX^T$ where X is the data matrix and D is diagonal.

After solving the subproblem (8), we have to search for a suitable stepsize. We apply an Armijo rule for line search [24], where we test the step size $\alpha = 2^0, 2^{-1}, \ldots$ until the following sufficient decrease condition is satisfied for a pre-specified $\sigma \in (0, 1)$ (typically $\sigma = 10^{-4}$):

$$F(\boldsymbol{\theta} + \alpha\boldsymbol{\Delta}) \leq F(\boldsymbol{\theta}) + \alpha\sigma\delta, \;\; \delta = \langle G, \boldsymbol{\Delta} \rangle + \sum_{r=1}^{k} \lambda_r \|\Theta_r + \alpha\boldsymbol{\Delta}^{(r)}\|_{\mathcal{A}_r} - \sum_{r=1}^{k} \lambda_r \|\boldsymbol{\theta}^{(r)}\|_{\mathcal{A}_r}. \tag{10}$$

3.2 Active Subspace Selection

Since the quadratic subproblem (8) contains a large number of variables, directly applying the above quadratic approximation framework is not efficient. In this subsection, we provide a general *active subspace selection* technique, which dramatically reduces the size of variables by exploiting the structure of regularizers. A similar method has been discussed in [12] for the ℓ_1 norm and in [11] for the nuclear norm, but it has not been generalized to all decomposable norms. Furthermore, a key point to note is that in this paper our active subspace selection is not only a heuristic, but comes with strong convergence guarantees that we derive in Section 4.

Given the current $\boldsymbol{\theta}$, our subspace selection approach partitions each $\boldsymbol{\theta}^{(r)}$ into $\mathcal{S}^{(r)}_{\text{fixed}}$ and $\mathcal{S}^{(r)}_{\text{free}} = (\mathcal{S}^{(r)}_{\text{fixed}})^{\perp}$ and then restricts the search space of the Newton direction in (8) within $\mathcal{S}^{(r)}_{\text{free}}$, which yields the following quadratic approximation problem:

$$[\boldsymbol{d}^{(1)}, \ldots, \boldsymbol{d}^{(k)}] = \operatorname*{argmin}_{\boldsymbol{\Delta}^{(1)} \in \mathcal{S}^{(1)}_{\text{free}}, \ldots, \boldsymbol{\Delta}^{(k)} \in \mathcal{S}^{(k)}_{\text{free}}} \bar{g}(\boldsymbol{\theta} + \boldsymbol{\Delta}) + \sum_{r=1}^{k} \lambda_r \|\boldsymbol{\theta}^{(r)} + \boldsymbol{\Delta}^{(r)}\|_{\mathcal{A}_r}. \tag{11}$$

Each group of parameter has its own fixed/free subspace, so we now focus on a single parameter component $\boldsymbol{\theta}^{(r)}$. An ideal subspace selection procedure would satisfy:

Property (I). Given the current iterate $\boldsymbol{\theta}$, any updates along directions in the fixed set, for instance as $\boldsymbol{\theta}^{(r)} \leftarrow \boldsymbol{\theta}^{(r)} + \boldsymbol{a},\ \boldsymbol{a} \in \mathcal{S}_{\text{fixed}}^{(r)}$, does not improve the objective function value.

Property (II). The subspace $\mathcal{S}_{\text{free}}$ converges to the support of the final solution in a finite number of iterations.

Suppose given the current iterate, we first do updates along directions in the fixed set, and then do updates along directions in the free set. Property (I) ensures that this is equivalent to ignoring updates along directions in the fixed set in this current iteration, and focusing on updates along the free set. As we will show in the next section, this property would suffice to ensure global convergence of our procedure. Property (II) will be used to derive the asymptotic quadratic convergence rate.

We will now discuss our active subspace selection strategy which will satisfy both properties above. Consider the parameter component $\boldsymbol{\theta}^{(r)}$, and its corresponding regularizer $\|\cdot\|_{\mathcal{A}_r}$. Based on the definition of decomposable norm in (2), there exists a subspace $\mathcal{T}_r$ where $\Pi_{\mathcal{T}_r}(\boldsymbol{\rho})$ is a fixed vector for any subgradient of $\|\cdot\|_{\mathcal{A}_r}$. The following proposition explores some properties of the sub-differential of the overall objective $F(\boldsymbol{\theta})$ in (1).

Proposition 1. *Consider any unit-norm vector $\boldsymbol{a}$, with $\|\boldsymbol{a}\|_{\mathcal{A}_r} = 1$, such that $\boldsymbol{a} \in \mathcal{T}_r^{\perp}$.*

(a) The inner-product of the sub-differential $\partial_{\boldsymbol{\theta}^{(r)}} F(\boldsymbol{\theta})$ with $\boldsymbol{a}$ satisfies:

$$\langle \boldsymbol{a}, \partial_{\boldsymbol{\theta}^{(r)}} F(\boldsymbol{\theta}) \rangle \in [\langle \boldsymbol{a}, G \rangle - \lambda_r, \langle \boldsymbol{a}, G \rangle + \lambda_r]. \tag{12}$$

(b) Suppose $|\langle \boldsymbol{a}, G \rangle| \leq \lambda_r$. Then, $0 \in \operatorname{argmin}_{\sigma} F(\boldsymbol{\theta} + \sigma \boldsymbol{a})$.

See Appendix 7.8 for the proof. Note that $G = \nabla \mathcal{L}(\bar{\boldsymbol{\theta}})$ denotes the gradient of $\mathcal{L}$. The proposition thus implies that if $|\langle \boldsymbol{a}, G \rangle| \leq \lambda_r$ and $\mathcal{S}_{\text{fixed}}^{(r)} \subset \mathcal{T}_r^{\perp}$ then Property (I) immediately follows. The difficulty is that the set $\{\boldsymbol{a} \mid |\langle \boldsymbol{a}, G \rangle| \leq \lambda_r\}$ is possibly hard to characterize, and even if we could characterize this set, it may not be amenable enough for the optimization solvers to leverage in order to provide a speedup. Therefore, we propose an alternative characterization of the fixed subspace:

Definition 1. *Let $\boldsymbol{\theta}^{(r)}$ be the current iterate, $\operatorname{prox}_{\lambda}^{(r)}$ be the proximal operator defined by*

$$\operatorname{prox}_{\lambda}^{(r)}(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{y}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 + \lambda \|\boldsymbol{y}\|_{\mathcal{A}_r},$$

and $\mathcal{T}_r(\boldsymbol{x})$ be the subspace for the decomposable norm (2) $\|\cdot\|_{\mathcal{A}_r}$ at point $\boldsymbol{x}$. We can define the fixed/free subset at $\boldsymbol{\theta}^{(r)}$ as:

$$\mathcal{S}_{\textit{fixed}}^{(r)} := [\mathcal{T}(\boldsymbol{\theta}^{(r)})]^{\perp} \cap [\mathcal{T}(\operatorname{prox}_{\lambda_r}^{(r)}(G))]^{\perp},\ \ \mathcal{S}_{\textit{free}}^{(r)} = {\mathcal{S}_{\textit{fixed}}^{(r)}}^{\perp}. \tag{13}$$

It can be shown that from the definition of the proximal operator, and Definition 1, it holds that $|\langle \boldsymbol{a}, G \rangle| < \lambda_r$, so that we would have local optimality in the direction $\boldsymbol{a}$ as before. We have the following proposition:

Proposition 2. *Let $\mathcal{S}_{\textit{fixed}}^{(r)}$ be the fixed subspace defined in Definition 1. We then have:*

$$0 = \operatorname*{argmin}_{\boldsymbol{\Delta}^{(r)} \in \mathcal{S}_{\textit{fixed}}^{(r)}} Q_{\mathcal{H}}([\mathbf{0}, \ldots, \mathbf{0}, \boldsymbol{\Delta}^{(r)}, \mathbf{0}, \ldots, \mathbf{0}]; \boldsymbol{\theta}).$$

We will prove that $\mathcal{S}_{\text{free}}$ as defined above converges to the final support in Section 4, as required in Property (II) above. We will now detail some examples of the fixed/free subsets defined above.

- For ℓ_1 regularization: $\mathcal{S}_{\text{fixed}} = \text{span}\{e_i \mid \theta_i = 0 \text{ and } |\nabla_i \mathcal{L}(\bar{\boldsymbol{\theta}})| \leq \lambda\}$ where e_i is the i^{th} canonical vector.
- For nuclear norm regularization: the selection scheme can be written as

$$\mathcal{S}_{\text{free}} = \{U_A M V_A^T \mid M \in \mathbb{R}^{k \times k}\}, \tag{14}$$

where $U_A = \text{span}(U, U_g), V_A = \text{span}(V, V_g)$, with $\Theta = U \Sigma V^T$ is the thin SVD of Θ and U_g, V_g are the left and right singular vectors of $\text{prox}_{\lambda}(\Theta - \nabla \mathcal{L}(\Theta))$. The proximal operator $\text{prox}_{\lambda}(\cdot)$ in this case corresponds to singular-value soft-thresholding, and can be computed by randomized SVD or the Lanczos algorithm.

• For group sparse regularization: in the $(1,2)$-group norm case, let S_G be the nonzero groups, then the fixed groups F_G can be defined by $F_G := \{i \mid i \notin S_G \text{ and } \|\nabla \mathcal{L}_{G_i}(\bar{\boldsymbol{\theta}})\| \leq \lambda\}$, and the free subspace will be

$$\mathcal{S}_{\text{free}} = \{\boldsymbol{\theta} \mid \boldsymbol{\theta}_i = 0 \; \forall i \in F_G\}. \tag{15}$$

In Figure 3 (in the appendix) that the active subspace selection can significantly improve the speed for the block coordinate descent algorithm [20].

Algorithm 1: QUIC & DIRTY: Quadratic Approximation Framework for Dirty Statistical Models

Input : Loss function $\mathcal{L}(\cdot)$, regularizers $\lambda_r \|\cdot\|_{\mathcal{A}_r}$ for $r = 1, \ldots, k$, and initial iterate $\boldsymbol{\theta}_0$.
Output: Sequence $\{\boldsymbol{\theta}_t\}$ such that $\{\bar{\boldsymbol{\theta}}_t\}$ converges to $\bar{\boldsymbol{\theta}}^\star$.

for $t = 0, 1, \ldots$ **do**
  Compute $\bar{\boldsymbol{\theta}}_t \leftarrow \sum_{r=1}^k \boldsymbol{\theta}_t^{(r)}$.
  Compute $\nabla \mathcal{L}(\bar{\boldsymbol{\theta}}_t)$.
  Compute $\mathcal{S}_{\text{free}}$ by (13).
  for *sweep* $= 1, \ldots, T_{outer}$ **do**
    for $r = 1, \ldots, k$ **do**
      Solve the subproblem (9) within $\mathcal{S}_{\text{free}}^{(r)}$.
      Update $\sum_{r=1}^k \nabla^2 \mathcal{L}(\bar{\boldsymbol{\theta}}_t) \boldsymbol{\Delta}^{(r)}$.
  Find the step size α by (10).
  $\boldsymbol{\theta}^{(r)} \leftarrow \boldsymbol{\theta}^{(r)} + \alpha \boldsymbol{\Delta}^{(r)}$ for all $r = 1, \ldots, k$.

4 Convergence

The recently developed theoretical analysis of proximal Newton methods [16, 21] cannot be directly applied because (1) we have the active subspace selection step, and (2) the Hessian matrix for each quadratic subproblem is *not* positive definite. We first prove the global convergence of our algorithm when the quadratic approximation subproblem (11) is solved exactly. Interestingly, in our proof we show that the active subspace selection can be modeled within the framework of the Block Coordinate Gradient Descent algorithm [24] with a carefully designed Hessian approximation, and by making this connection we are able to prove global convergence.

Theorem 1. *Suppose $\mathcal{L}(\cdot)$ is convex (may not be strongly convex), and the quadratic subproblem* (8) *at each iteration is solved exactly, Algorithm 1 converges to the optimal solution.*

The proof is in Appendix 7.1. Next we consider the case that $\mathcal{L}(\bar{\boldsymbol{\theta}})$ is strongly convex. Note that even when $\mathcal{L}(\bar{\boldsymbol{\theta}})$ is strongly convex with respect to $\bar{\boldsymbol{\theta}}$, $\mathcal{L}(\sum_{r=1}^k \boldsymbol{\theta}^{(r)})$ will not be strongly convex in $\boldsymbol{\theta}$ (if $k > 1$) and there may exist more than one optimal solution. However, we show that all solutions give the same $\bar{\boldsymbol{\theta}} := \sum_{r=1}^k \boldsymbol{\theta}^{(r)}$.

Lemma 2. *Assume $\mathcal{L}(\cdot)$ is strongly convex, and $\{\boldsymbol{x}^{(r)}\}_{r=1}^k, \{\boldsymbol{y}^{(r)}\}_{r=1}^k$ are two optimal solutions of* (1)*, then $\sum_{r=1}^k \boldsymbol{x}^{(r)} = \sum_{r=1}^k \boldsymbol{y}^{(r)}$.*

The proof is in Appendix 7.2. Next, we show that $\mathcal{S}_{\text{free}}^{(r)}$ (from Definition 1) will converge to the final support $\bar{\mathcal{T}}^{(r)}$ for each parameter set $r = 1, \ldots, k$. Let $\bar{\boldsymbol{\theta}}^\star$ be the global minimizer (which is unique as shown in Lemma 2), and assume that we have

$$\|\Pi_{(\bar{\mathcal{T}}^{(r)})^\perp} \left(\nabla \mathcal{L}(\bar{\boldsymbol{\theta}}^\star)\right)\|_{\mathcal{A}_r}^* < \lambda_r \;\; \forall r = 1, \ldots, k. \tag{16}$$

This is the generalization of the assumption used in earlier literature [12] where only ℓ_1 regularization was considered. The condition is similar to strict complementary in linear programming.

Theorem 2. *If $\mathcal{L}(\cdot)$ is strongly convex and assumption* (16) *holds, then there exists a finite $T > 0$ such that $\mathcal{S}_{free}^{(r)} = \bar{\mathcal{T}}^{(r)} \;\; \forall r = 1, \ldots, k$ after $t > T$ iterations.*

The proof is in Appendix 7.3. Next we show that our algorithm has an asymptotic quadratic convergence rate (the proof is in Appendix 7.4).

Theorem 3. *Assume that $\nabla^2 \mathcal{L}(\cdot)$ is Lipschitz continuous, and assumption* (16) *holds. If at each iteration the quadratic subproblem* (8) *is solved exactly, and $\mathcal{L}(\cdot)$ is strongly convex, then our algorithm converges with asymptotic quadratic convergence rate.*

5 Applications

We demonstrate that our algorithm is extremely efficient for two applications: Gaussian Markov Random Fields (GMRF) with latent variables (with sparse + low rank structure) and multi-task learning problems (with sparse + group sparse structure).

5.1 GMRF with Latent Variables

We first apply our algorithm to solve the latent feature GMRF structure learning problem in eq (4), where $S \in \mathbb{R}^{p\times p}$ is the sparse part, $L \in \mathbb{R}^{p\times p}$ is the low-rank part, and we require $L = L^T \succeq 0, S = S^T$ and $Y = S - L \succ 0$ (i.e. $\boldsymbol{\theta}^{(2)} = -L$). In this case, $\mathcal{L}(Y) = -\log\det(Y) + \langle \Sigma, Y\rangle$, hence

$$\nabla^2 \mathcal{L}(Y) = Y^{-1} \otimes Y^{-1}, \text{ and } \nabla \mathcal{L}(Y) = \Sigma - Y^{-1}. \tag{17}$$

Active Subspace. For the sparse part, the free subspace is a subset of indices $\{(i,j) \mid S_{ij} \neq 0 \text{ or } |\nabla_{ij}\mathcal{L}(Y)| \geq \lambda\}$. For the low-rank part, the free subspace can be presented as $\{U_A M V_A^T \mid M \in \mathbb{R}^{k\times k}\}$ where U_A and V_A are defined in (14).

Updating Δ_L. To solve the quadratic subproblem (11), first we discuss how to update Δ_L using subspace selection. The subproblem is

$$\min_{\Delta_L = U\Delta_D U^T : L+\Delta_L \succeq 0} \frac{1}{2}\operatorname{trace}(\Delta_L Y^{-1}\Delta_L Y^{-1}) + \operatorname{trace}((Y^{-1} - \Sigma - Y^{-1}\Delta_S Y^{-1})\Delta_L) + \lambda_L \|L + \Delta_L\|_*,$$

and since Δ_L is constrained to be a perturbation of $L = U_A M U_A^T$ so that we can write $\Delta_L = U_A \Delta_M U_A^T$, and the subproblem becomes

$$\min_{\Delta_M : M + \Delta_M \succeq 0} \frac{1}{2}\operatorname{trace}(\bar{Y}\Delta_M \bar{Y}\Delta_M) + \operatorname{trace}(\bar{\Sigma}\Delta_M) + \lambda_L \operatorname{trace}(M + \Delta_M) := q(\Delta_M), \tag{18}$$

where $\bar{Y} := U_A^T Y^{-1} U_A$ and $\bar{\Sigma} := U_A^T (Y^{-1} - \Sigma - Y^{-1}\Delta_S Y^{-1}) U_A$. Therefore the subproblem (18) becomes a $k \times k$ dimensional problem where $k \ll p$.

To solve (18), we first check if the closed form solution exists. Note that $\nabla q(\Delta_M) = \bar{Y}\Delta_M\bar{Y} + \bar{\Sigma} + \lambda_L I$, thus the minimizer is $\Delta_M = -\bar{Y}^{-1}(\bar{\Sigma} + \lambda_L I)\bar{Y}^{-1}$ if $M + \Delta_M \succeq 0$. If not, we solve the subproblem by the projected gradient descent method, where each step only requires $O(k^2)$ time.

Updating Δ_S. The subproblem with respect to Δ_S can be written as

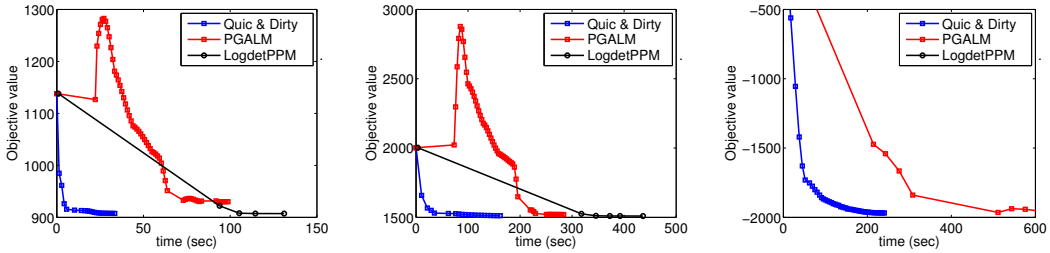
$$\min_{\Delta_S} \frac{1}{2}\operatorname{vec}(\Delta_S)^T (Y^{-1} \otimes Y^{-1}) \operatorname{vec}(\Delta_S) + \operatorname{trace}((\Sigma - Y^{-1} - Y^{-1}(\Delta_L)Y^{-1})\Delta_S) + \lambda_S \|S + \Delta_S\|_1,$$

In our implementation we apply the same coordinate descent procedure proposed in QUIC [12] to solve this subproblem.

Results. We compare our algorithm with two state-of-the-art software packages. The LogdetPPA algorithm was proposed in [26] and used in [5] to solve (4). The PGALM algorithm was proposed in [17]. We run our algorithm on three gene expression datasets: the ER dataset ($p = 692$), the Leukemia dataset ($p = 1255$), and a subset of the Rosetta dataset ($p = 2000$)[1] For the parameters, we use $\lambda_S = 0.5, \lambda_L = 50$ for the ER and Leukemia datasets, which give us low-rank and sparse results. For the Rosetta dataset, we use the parameters suggested in LogdetPPA, with $\lambda_S = 0.0313, \lambda_L = 0.1565$. The results in Figure 1 shows that our algorithm is more than 10 times faster than other algorithms. Note that in the beginning PGALM tends to produce infeasible solutions (L or $S - L$ is not positive definite), which is not plotted in the figures.

Our proximal Newton framework has two algorithmic components: the quadratic approximation, and our active subspace selection. From Figure 1 we can observe that although our algorithm is a Newton-like method, the time cost for each iteration is similar or even cheaper than other first order methods. The reason is (1) we take advantage from active selection, and (2) the problem has a special structure of the Hessian (17), where computing it is no more expensive than the gradient. To delineate the contribution of the quadratic approximation to the gain in speed of convergence, we further compare our algorithm to an alternating minimization approach for solving (4), together with our active subspace selection. Such an alternating minimization approach would iteratively fix one of S, L, and update the other; we defer detailed algorithmic and implementation details to Appendix 7.6 for reasons of space. The results show that by using the quadratic approximation, we get a much faster convergence rate (see Figure 2 in Appendix 7.6).

[1]The full dataset has $p = 6316$ but the other methods cannot solve this size problem.



(a) ER dataset (b) Leukemia dataset (c) Rosetta dataset

Figure 1: Comparison of algorithms on the latent feature GMRF problem using gene expression datasets. Our algorithm is much faster than PGALM and LogdetPPA.

Table 1: The comparisons on multi-task problems.

dataset	number of training data	relative error	Dirty Models (sparse + group sparse)			Other Models	
			QUIC & DIRTY	proximal gradient	ADMM	Lasso	Group Lasso
USPS	100	10^{-1}	8.3% / **0.42s**	8.5% / 1.8s	8.3% / 1.3	10.27%	8.36%
	100	10^{-4}	7.47% / **0.75s**	7.49% / 10.8s	7.47% / 4.5s		
	400	10^{-1}	2.92% / **1.01s**	2.9% / 9.4s	3.0% / 3.6s	4.87%	2.93%
	400	10^{-4}	2.5% / **1.55s**	2.5% / 35.8	2.5% / 11.0s		
RCV1	1000	10^{-1}	18.91% / **10.5s**	18.5%/47s	18.9% / 23.8s	22.67%	20.8%
	1000	10^{-4}	18.45% / **23.1s**	18.49% / 430.8s	18.5% / 259s		
	5000	10^{-1}	10.54% / **42s**	10.8% / 541s	10.6% / 281s	13.67%	12.25%
	5000	10^{-4}	10.27% / **87s**	10.27% / 2254s	10.27% / 1191s		

5.2 Multiple-task learning with superposition-structured regularizers

Next we solve the multi-task learning problem (5) where the parameter is a sparse matrix $S \in \mathbb{R}^{d\times k}$ and a group sparse matrix $B \in \mathbb{R}^{d\times k}$. Instead of using the square loss (as in [15]), we consider the logistic loss $\ell_{\text{logistic}}(y, a) = \log(1 + e^{-ya})$, which gives better performance as seen by comparing Table 1 to results in [15]. Here the Hessian matrix has a special structure again: $H = XDX^T$ where X is the data matrix and D is the diagonal matrix, and in Appendix 7.7 we have a detail description of how to applying our algorithm to solve this problem.

Results. We follow [15] and transform multi-class problems into multi-task problems. For a multiclass dataset with k classes and n samples, for each $r = 1, \ldots, k$, we generate $\boldsymbol{y}^r \in \{0, 1\}^n$ to be the vector such that $y_i^{(k)} = 1$ if and only if the i-th sample is in class r. Our first dataset is the USPS dataset which was first collected in [25] and subsequently widely used in multi-task papers. On this dataset, the use of several regularizers is crucial for good performance. For example, [15] demonstrates that on USPS, using lasso and group lasso regularizations together outperforms models with a single regularizer. However, they only consider the squared loss in their paper, whereas we consider a logistic loss which leads to better performance. For example, we get 7.47% error rate using 100 samples in USPS dataset, while using the squared loss the error rate is 10.8% [15]. Our second dataset is a larger document dataset RCV1 downloaded from LIBSVM Data, which has 53 classes and 47,236 features. We show that our algorithm is much faster than other algorithms on both datasets, especially on RCV1 where we are more than 20 times faster than proximal gradient descent. Here our subspace selection techniques works well because we expect that the active subspace at the true solution is small.

6 Acknowledgements

This research was supported by NSF grants CCF-1320746 and CCF-1117055. C.-J.H also acknowledges support from an IBM PhD fellowship. P.R. acknowledges the support of ARO via W911NF-12-1-0390 and NSF via IIS-1149803, IIS-1447574, and DMS-1264033. S.R.B. was supported by an IBM Research Goldstine Postdoctoral Fellowship while the work was performed.

References

[1] A. Agarwal, S. Negahban, and M. J. Wainwright. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *Annals if Statistics*, 40(2):1171–1197, 2012.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[3] E. Candes and B. Recht. Simple bounds for recovering low-complexity models. *Mathemetical Programming*, 2012.

[4] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. Assoc. Comput. Mach.*, 58(3):1–37, 2011.

[5] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky. Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 2012.

[6] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *Siam J. Optim*, 21(2):572–596, 2011.

[7] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis. Low-rank matrix recovery from errors and erasures. *IEEE Transactions on Information Theory*, 59(7):4324–4337, 2013.

[8] M. Collins, S. Dasgupta, and R. E. Schapire. A generalization of principal component analysis to the exponential family. In *NIPS*, 2012.

[9] Q. T. Dinh, A. Kyrillidis, and V. Cevher. An inexact proximal path-following algorithm for constrained convex minimization. *arxiv:1311.1756*, 2013.

[10] C.-J. Hsieh, I. S. Dhillon, P. Ravikumar, and A. Banerjee. A divide-and-conquer method for sparse inverse covariance estimation. In *NIPS*, 2012.

[11] C.-J. Hsieh and P. A. Olsen. Nuclear norm minimization via active subspace selection. In *ICML*, 2014.

[12] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar. Sparse inverse covariance matrix estimation using quadratic approximation. In *NIPS*, 2011.

[13] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, P. Ravikumar, and R. A. Poldrack. BIG & QUIC: Sparse inverse covariance estimation for a million variables. In *NIPS*, 2013.

[14] D. Hsu, S. M. Kakade, and T. Zhang. Robust matrix decomposition with sparse corruptions. *IEEE Trans. Inform. Theory*, 57:7221–7234, 2011.

[15] A. Jalali, P. Ravikumar, S. Sanghavi, and C. Ruan. A dirty model for multi-task learning. In *NIPS*, 2010.

[16] J. D. Lee, Y. Sun, and M. A. Saunders. Proximal Newton-type methods for convex optimization. In *NIPS*, 2012.

[17] S. Ma, L. Xue, and H. Zou. Alternating direction methods for latent variable Gaussian graphical model selection. *Neural Computation*, 25(8):2172–2198, 2013.

[18] S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

[19] P. Olsen, F. Oztoprak, J. Nocedal, and S. Rennie. Newton-like methods for sparse inverse covariance estimation. In *NIPS*, 2012.

[20] Z. Qin, K. Scheinberg, and D. Goldfarb. Efficient block-coordinate descent algorithm for the group lasso. *Mathematical Programming Computation*, 2013.

[21] K. Scheinberg and X. Tang. Practical inexact proximal quasi-newton method with global complexity analysis. *arxiv:1311.6547*, 2014.

[22] A. Tewari, P. Ravikumar, and I. Dhillon. Greedy algorithms for structurally constrained high dimensional problems. In *NIPS*, 2011.

[23] K.-C. Toh, P. Tseng, and S. Yun. A block coordinate gradient descent method for regularized convex separable optimization and covariance selection. *Mathemetical Programming*, 129:331–355, 2011.

[24] P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117:387–423, 2007.

[25] M. van Breukelen, R. P. W. Duin, D. M. J. Tax, and J. E. den Hartog. Handwritten digit recognition by combined classifiers. *Kybernetika*, 34(4):381–386, 1998.

[26] C. Wang, D. Sun, and K.-C. Toh. Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm. *SIAM J. Optimization*, 20:2994–3013, 2010.

[27] E. Yang and P. Ravikumar. Dirty statistical models. In *NIPS*, 2013.

[28] E.-H. Yen, C.-J. Hsieh, P. Ravikumar, and I. S. Dhillon. Constant nullspace strong convexity and fast convergence of proximal methods under high-dimensional settings. In *NIPS*, 2014.

[29] G.-X. Yuan, C.-H. Ho, and C.-J. Lin. An improved GLMNET for L1-regularized logistic regression. *JMLR*, 13:1999–2030, 2012.

7 Appendix

7.1 Proof of Theorem 1

Proof. There are two main difficulties in proving the convergence of our algorithm, and none of them is addressed in previous works. First, the Hessian matrix $\mathcal{H}$ is a block-structured matrix as shown in (7), and unfortunately it is low-rank. Second, we have to show the convergence with the active set selection technique.

Let $H(\boldsymbol{\theta}) \in \mathcal{R}^{n\times n}$ be the Hessian $\nabla^2\mathcal{L}(\bar{\boldsymbol{\theta}})$ when $\mathcal{L}(\cdot)$ is strongly convex. When it is not, as discussed in Section 3, we set $H = \nabla^2\mathcal{L}(\bar{\boldsymbol{\theta}}) + \epsilon I$ for a small constant $\epsilon > 0$. Let $\boldsymbol{d} \in \mathbb{R}^{n^2k}$ denotes the minimizer of the quadratic subproblem (8). By definition, we can easily observe that $\sum_{r=1}^{k} \boldsymbol{d}^{(r)} = \boldsymbol{y}$ where

$$\boldsymbol{y}^\star := \operatorname*{argmin}_{\boldsymbol{y}} \boldsymbol{y}^T H(\boldsymbol{\theta})\boldsymbol{y} + \langle \boldsymbol{y}, \nabla\mathcal{L}(\bar{\boldsymbol{\theta}})\rangle + \mathcal{W}(\boldsymbol{y}), \tag{19}$$

where $\mathcal{W}(\boldsymbol{y}) = \min_{\boldsymbol{y}^{(1)},\ldots,\boldsymbol{y}^{(k)}:\sum_{r=1}^{k}\boldsymbol{y}^{(r)}=\boldsymbol{y}} \lambda_r\|\boldsymbol{y}^{(r)}\|_{\mathcal{A}_r}$. Now (19) has a strongly convex Hessian, and thus if $\mathcal{W}(\boldsymbol{y})$ is convex then Theorem 3.1 in [16] can be used to show that the algorithm converges, and thus $F(\sum_{i=1}^{r}\boldsymbol{\theta}^{(r)})$ converges (without active subspace selection).

To show $\mathcal{W}$ is convex, assume we have $\boldsymbol{a}, \boldsymbol{b}$ and $\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(k)}$ are decomposition of $\boldsymbol{a}$ that attains the minimizer of $\mathcal{W}(\boldsymbol{a})$. By definition we have

$$\begin{aligned}
\mathcal{W}(\alpha\boldsymbol{a} + (1-\alpha)\boldsymbol{b}) &\le \sum_{r=1}^{k} \lambda_r\|\boldsymbol{a}^{(r)} + \boldsymbol{b}^{(r)}\|_{\mathcal{A}_r} \\
&\le \sum_{r=1}^{k} \lambda_r(\alpha\|\boldsymbol{a}^{(r)}\|_{\mathcal{A}_r} + (1-\alpha)\|\boldsymbol{b}^{(r)}\|_{\mathcal{A}_r}) \\
&= \alpha\mathcal{W}(\boldsymbol{a}) + (1-\alpha)\mathcal{W}(\boldsymbol{b}).
\end{aligned}$$

Thus $\mathcal{W}$ is convex, and if we solve each quadratic approximation exactly without active subspace selection, our algorithm converges to the global optimum.

Next we discuss the convergence of our algorithm with active subspace selection. [12] has shown the convergence of active set selection under ℓ_1 regularization, but here the situation is different – we can have infinite many atoms, as in group lasso or nuclear norm cases, so the original proof cannot be applied. To analyze the active subspace selection technique, we will use the convergence proof for Block Coordinate Gradient Descent (BCGD) in [23]. To begin, we give a quick review of the Block Coordinate Gradient Descent (BCGD) algorithm discussed in [23].

BCGD is proposed to solve the composite functions with the following form:

$$\operatorname*{argmin}_{\boldsymbol{x}} F(\boldsymbol{x}) := f(\boldsymbol{x}) + P(\boldsymbol{x}),$$

where $P(\boldsymbol{x})$ is a convex and separable function. Here we consider $\boldsymbol{x}$ to be a p dimensional vector, but in general $\boldsymbol{x}$ can be in any space. At the t-th iteration, BCGD chooses a subset $\mathcal{J}_t$ and compute the descent direction by

$$\boldsymbol{d}_t = \boldsymbol{d}_{H_t}^{\mathcal{J}_t}(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{d}} \left\{ \langle \nabla f(\boldsymbol{x}), \boldsymbol{d}\rangle + \frac{1}{2}\boldsymbol{d}^T H_t \boldsymbol{d} + P(\boldsymbol{x}+\boldsymbol{d}) \mid d_i = 0, \forall i \notin \mathcal{J} \right\} \equiv \operatorname*{argmin}_{\boldsymbol{d}} Q_{H_t}^{\mathcal{J}_t}(\boldsymbol{d}). \tag{20}$$

After computing the descent direction $\boldsymbol{d}_H^{\mathcal{J}_t}$, a line search procedure is used to find the descent direction, where the largest α_t is chosed by searching over $1, 1/2, 1/4, ...$ satisfying

$$F(\boldsymbol{x}_t + \alpha\boldsymbol{d}_t) \le F(\boldsymbol{x}_t) + \alpha_t\sigma\Delta_t, \text{ where } \Delta_t = \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{d}_t\rangle + P(\boldsymbol{x}_t + \boldsymbol{d}_t) - P(\boldsymbol{x}_t),$$

and $\sigma \in (0,1)$ is any constant. The line search condition is exactly the same with ours in (10).

It is shown in Theorem 3.1 of [23] that when J_t is selected in a cyclic order covering all the indexes, than BCGD converges to the global optimum for any convex function $f(X)$.

To proof this theorem, we want to show the equivalence between our algorithm and BCGD with one block (BCGD-1block). We first prove the convergence for the case where we only have one regularizer, thus the problem is

$$\operatorname*{argmin}_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}) + \lambda\|\boldsymbol{x}\|_{\mathcal{A}}.$$

The key idea is to carefully define the matrix H_t at each iteration, according the our subspace selection trick. Note that in (20) the matrix H_t can be any positive definite matrix instead of Hessian. At each iteration of Algorithm 1, assume the fixed and free subspace defined in (13) are $\mathcal{S}_{\text{fixed}}$ and $\mathcal{S}_{\text{free}}$. Assume $\dim(\mathcal{S}_{\text{free}}) = q$ and $\dim(\mathcal{S}_{\text{fixed}}) = n - q$, $R = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_q]$ are the orthogonal basis for $\mathcal{S}_{\text{free}}$, then we can construct $\tilde{H}_t$ by

$$\tilde{H}_t = R(R^T H_t R)R^T + R^{\perp}(R^{\perp})^T. \tag{21}$$

It is easy to see that $\tilde{H}_t$ is positive definite if the real Hessian $\nabla^2 f(\boldsymbol{x})$ is positive definite. Using $\tilde{H}_t$ as the Hessian in (20), we first consider $\boldsymbol{d}_{\text{free}}$ to be the minimizer of the following problem:

$$\boldsymbol{d}_{\text{free}} = \underset{\Delta_{\text{free}} \in \mathcal{S}_{\text{free}}}{\operatorname{argmin}} \left\{ \langle \nabla f(\boldsymbol{x}), \Delta_{\text{free}} \rangle + \frac{1}{2} {\Delta_{\text{free}}}^T \tilde{H}_t \Delta_{\text{free}} + P(\boldsymbol{x} + \Delta_{\text{free}}) \right\},$$

which is the quadratic subproblem (20) within the subspace $\mathcal{S}_{\text{free}}$. Next we show that $\boldsymbol{d}_{\text{free}}$ is indeed the optimizer of the whole quadratic subproblem (20) with the original Hessian H_t. To show this, taking derivative of (20) with respect to a $\boldsymbol{a} \in \mathcal{S}_{\text{fixed}}$, the subgradient will be

$$\partial_{\boldsymbol{a}} Q_{\tilde{H}_t}(\boldsymbol{d}_{\text{free}}) = \langle \nabla \mathcal{L}(\boldsymbol{x}), \boldsymbol{a} \rangle + \boldsymbol{a}^T \tilde{H}_t \boldsymbol{d}_{\text{free}} + \partial_{\boldsymbol{a}} P(\boldsymbol{x} + \boldsymbol{d}_{\text{free}}).$$

By the definition of $\tilde{H}_t$ in (21), since $\boldsymbol{a} \in \text{span}(R^{\perp})$ and $\boldsymbol{d}_{\text{free}} \in \text{span})(R)$, we have $\boldsymbol{a}^T \tilde{H}_t \boldsymbol{d}_{\text{free}} = 0$. Also, since $\boldsymbol{a} \in \mathcal{S}_{\text{fixed}}$, $\langle \boldsymbol{x} + \boldsymbol{d}_{\text{free}}, \boldsymbol{a} \rangle = 0$, thus $\partial_{\boldsymbol{a}} P(\boldsymbol{x} + \boldsymbol{d}_{\text{free}}) = [-\lambda, \lambda]$. Also, since $\boldsymbol{a} \in \mathcal{S}_{\text{fixed}}$, $|\langle \nabla f(\boldsymbol{x}), \boldsymbol{a} \rangle| < \lambda$. Therefore, we have

$$0 \in \partial_{\boldsymbol{a}} Q_{\tilde{H}_t}(\boldsymbol{d}_{\text{free}}), \ \ \forall \boldsymbol{a} \in S_{\text{fixed}},$$

also, since $\boldsymbol{d}_{\text{free}}$ is the optimal solution in $\mathcal{S}_{\text{free}}$, the projection of subgradient to $\boldsymbol{a} \in \mathcal{S}_{\text{free}}$ is 0. Therefore $\boldsymbol{d}_{\text{free}}$ is the minimizer of $Q_{\tilde{H}_t}$.

Based on the above discussion, our algorithm that computing the generalized Newton direction in free subspace $\mathcal{S}_{\text{free}}$ is equivalent to another BCGD-1block algorithm with $\tilde{H}_t$ as the approximated Hessian matrix. Therefore based on Theorem 3.1 of [23], our algorithm converges to the global optimum.

When there are more than one set of parameters, i.e., we want to solve (1) with parameter sets $\boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(k)}$. In this case, the Hessian matrix is a kn by kn matrix. For each parameter set, we will select ${\mathcal{S}_{\text{free}}}^{(i)}$ and ${\mathcal{S}_{\text{fixed}}}^{(i)}$, and only update on ${\mathcal{S}_{\text{free}}}^{(i)}$. To prove the convergence, similar to the above arguments, we can construct the approximate Hessian $\tilde{H}_t$ and show the equivalence between BCGD-block1 with $\tilde{H}_t$ and our algorithm. $\tilde{H}_t$ can be divided into k^2 blocks, each one is a n by n matrix $\tilde{H}_t^{(i,j)}$, where

$$\tilde{H}_t^{(i,j)} = R^{(i)}(R^{(i)})^T H R^{(j)}(R^{(j)})^T + (R^{(i)})^{\perp}((R^{(j)})^{\perp})^T,$$

where $H = \nabla^2 \mathcal{L}(\sum_{i=1}^{k} \boldsymbol{\theta}^{(r)})$, $R^{(i)}$ is the basis of ${\mathcal{S}_{\text{free}}}^{(i)}$, and $(R^{(i)})^{\perp}$ is the basis of ${\mathcal{S}_{\text{fixed}}}^{(i)}$. Assume $\boldsymbol{d}_{\text{free}} \in \mathcal{R}^{nk}$ is the solution of (20) with the constraint that projection to ${\mathcal{S}_{\text{fixed}}}^{(i)}$ is 0 for all i, then it is the solution for both BCGD-1block with $\tilde{H}_t$ and also the update direction produced by Algorithm 1. Also,

$$\boldsymbol{a} \tilde{H}_t \boldsymbol{d}_{\text{free}} = 0 \ \ \forall \boldsymbol{a} \in \{{\mathcal{S}_{\text{fixed}}}^{(1)}, \ldots, {\mathcal{S}_{\text{fixed}}}^{(k)}\},$$

therefore similar to the previous arguments we can show that

$$0 \in \partial_{\boldsymbol{a}} Q_{\tilde{H}_t}(\boldsymbol{d}_{\text{free}}), \ \forall \boldsymbol{a} \in \{{\mathcal{S}_{\text{fixed}}}^{(1)}, \ldots, {\mathcal{S}_{\text{fixed}}}^{(k)}\},$$

which indicates $\boldsymbol{d}_{\text{free}}$ is the minimizer of $Q_{\tilde{H}_t}(\boldsymbol{d})$. Then we can see that the BCGD-1block algorithm with $\tilde{H}_t$ as the Hessian matrix is equivalent to Algorithm 1 when each quadratic subproblem is solved exactly, therefore our algorithm converges to the global optimum of (1) □

7.2 Proof of Lemma 2

Proof. First, since $\mathcal{L}(\cdot)$ is strongly convex,

$$\langle \nabla \mathcal{L}(\bar{\boldsymbol{x}}) - \nabla \mathcal{L}(\bar{\boldsymbol{y}}), \bar{\boldsymbol{x}} - \bar{\boldsymbol{y}} \rangle \geq \eta \|\bar{\boldsymbol{x}} - \bar{\boldsymbol{y}}\|^2. \tag{22}$$

Next, by the optimal condition we know

$$-\nabla\mathcal{L}(\bar{\boldsymbol{x}}) \in \partial\|\boldsymbol{x}^{(r)}\|_{\mathcal{A}_r},\ \forall r = 1,\ldots,k$$
$$-\nabla\mathcal{L}(\bar{\boldsymbol{y}}) \in \partial\|\boldsymbol{y}^{(r)}\|_{\mathcal{A}_r},\ \forall r = 1,\ldots,k.$$

By the convexity for each $\|\cdot\|_{\mathcal{A}_r}$,

$$\langle -\nabla\mathcal{L}(\bar{\boldsymbol{x}}) + \nabla\mathcal{L}(\bar{\boldsymbol{y}}), \boldsymbol{x}^{(r)} - \boldsymbol{y}^{(r)}\rangle \geq 0,\ \forall r = 1,..,k.$$

Summing r from 1 to k we get $\langle -\nabla\mathcal{L}(\bar{\boldsymbol{x}}) + \nabla\mathcal{L}(\bar{\boldsymbol{y}}), \bar{\boldsymbol{x}} - \bar{\boldsymbol{y}}\rangle \geq 0$, and combined with (22) we have $\|\bar{\boldsymbol{x}} - \bar{\boldsymbol{y}}\| = 0$. □

7.3 Proof of Theorem 2

Proof. Since assumption (16) holds, there exists a positive constant ϵ such that

$$\|\Pi_{(\bar{\mathcal{T}}^{(r)})^\perp}\left(\nabla\mathcal{L}(\bar{\boldsymbol{\theta}}^\star)\right)\|^*_{\mathcal{A}_r} < \lambda_r - \epsilon\ \ \forall r = 1,\ldots,k. \tag{23}$$

We first focus on showing that that $\mathcal{S}_{\text{fixed}}$ will be eventually equal to $(\mathcal{T}^{(r)})^\perp$. Focus on one of the $(\mathcal{T}^{(r)})^\perp$, for any unit vector (in terms of the $\|\cdot\|_{\mathcal{A}_r}$ norm) $\boldsymbol{a} \in (\mathcal{T}^{(r)})^\perp$, we have

$$|\langle \boldsymbol{a}, \nabla\mathcal{L}(\bar{\boldsymbol{\theta}}^\star)\rangle| < \lambda_r - \epsilon. \tag{24}$$

Since the sequence generated by our algorithm converges to the global optimum (as proved in Theorem 1), there exists a T such that

$$\|\nabla\mathcal{L}(\bar{\boldsymbol{\theta}}_t) - \nabla\mathcal{L}(\bar{\boldsymbol{\theta}}^\star)\|^*_{\mathcal{A}_r} < \epsilon,$$

combining with (24) we have

$$\begin{aligned}|\langle \boldsymbol{a}, \nabla\mathcal{L}(\bar{\boldsymbol{\theta}}_t)\rangle| &\leq |\langle \boldsymbol{a}, \nabla\mathcal{L}(\bar{\boldsymbol{\theta}}^*)\rangle| + |\langle \boldsymbol{a}, \nabla\mathcal{L}(\bar{\boldsymbol{\theta}}_t) - \nabla\mathcal{L}(\bar{\boldsymbol{\theta}}^*)\rangle| \\ &< \lambda_r - \epsilon + \|\nabla\mathcal{L}(\bar{\boldsymbol{\theta}}_t) - \nabla\mathcal{L}(\bar{\boldsymbol{\theta}}^*)\|^*_{\mathcal{A}_r} \\ &< \lambda_r \end{aligned} \tag{25}$$

for all $t > T$. Now we consider two cases:

1. If $\langle \boldsymbol{a}, \boldsymbol{\theta}_{t-1}\rangle \neq 0$, then $\boldsymbol{a} \notin {\mathcal{S}_{\text{free}}}^{(r)}$ at the t-th iteration. Since we assume subproblems are exactly solved, and $\boldsymbol{a} \in {\mathcal{S}_{\text{free}}}^{(r)}$, by the optimality condition $|\langle\nabla\mathcal{L}(\bar{\boldsymbol{\theta}}_t)| < \lambda$ implies that $\langle \boldsymbol{\theta}_t, \boldsymbol{a}\rangle = 0$.
2. If $\langle \boldsymbol{a}, \boldsymbol{\theta}_{t-1}\rangle = 0$, combined with (25) we have $\langle \boldsymbol{\theta}_t, \boldsymbol{a}\rangle = 0$.

Therefore, for all $\boldsymbol{a} \in (\mathcal{T}^{(r)})^\perp$, we have $\langle \boldsymbol{\theta}_t, \boldsymbol{a}\rangle = 0$, which implies $(\mathcal{T}^{(r)})^\perp \subset \mathcal{S}_{\text{fixed}}$. On the other hand, by definition $(\mathcal{T}^{(r)}) \cap \mathcal{S}_{\text{fixed}} = \phi$, thus $\mathcal{T}^{(r)} = \mathcal{S}_{\text{free}}$ and $(\mathcal{T}_{(r)})^\perp = \mathcal{S}_{\text{fixed}}$. □

7.4 Proof of Theorem 3

Proof. As shown in Theorem 2, there exists a T such that $\mathcal{S}_{\text{free}} = \text{span}(\{\boldsymbol{a} \mid \langle \boldsymbol{a}, \Theta^\star\rangle\})$ after $t > T$.

Next we show that after finite iterations the line search step size will be 1. For simplicity, let $\bar{\mathcal{L}}(\boldsymbol{\theta}) = \mathcal{L}(\sum_{r=1}^k \boldsymbol{\theta}^{(r)})$ and $\bar{\mathcal{R}}(\boldsymbol{\theta}) = \sum_{r=1}^k \|\boldsymbol{\theta}_r\|_{\mathcal{A}_r}$, and $F(\boldsymbol{\theta}) = \bar{\mathcal{L}}(\boldsymbol{\theta}) + \bar{\mathcal{R}}(\boldsymbol{\theta})$. Since $\nabla^2\mathcal{L}(\bar{\boldsymbol{\theta}})$ is Lipschitz continuous, we have

$$\bar{\mathcal{L}}(\boldsymbol{\theta} + \boldsymbol{d}) \leq \bar{\mathcal{L}}(\boldsymbol{\theta}) + \langle \nabla\bar{\mathcal{L}}(\boldsymbol{\theta}), \boldsymbol{d}\rangle + \frac{1}{2}\boldsymbol{d}^T\nabla^2\bar{\mathcal{L}}(\boldsymbol{\theta})\boldsymbol{d} + \frac{1}{6}\eta\|\boldsymbol{d}\|^3.$$

Plug in into the objective function we have

$$\begin{aligned}F(\boldsymbol{\theta} + \boldsymbol{d}) &\leq \bar{\mathcal{L}}(\boldsymbol{\theta}) + \bar{\mathcal{R}}(\boldsymbol{\theta}) + \langle \nabla\bar{\mathcal{L}}(\boldsymbol{\theta}), \boldsymbol{d}\rangle + \\ &\quad \frac{1}{2}\boldsymbol{d}^T\nabla^2\bar{\mathcal{L}}(\boldsymbol{\theta})\boldsymbol{d} + \frac{1}{6}\eta\|\boldsymbol{d}\|^3 + \bar{\mathcal{R}}(\boldsymbol{\theta} + \boldsymbol{d}) - \bar{\mathcal{R}}(\boldsymbol{\theta}) \\ &\leq F(\boldsymbol{\theta}) + \delta + \frac{1}{2}\boldsymbol{d}^T\nabla^2\bar{\mathcal{L}}(\boldsymbol{\theta})\boldsymbol{d} + \frac{1}{6}\eta\|\boldsymbol{d}\|^3 \end{aligned} \tag{26}$$

To further bound (26), we first show the following Lemma:

Lemma 3. *Let $\boldsymbol{d}^\star$ be the optimal solution of* (20)*, then*

$$\delta = \langle \nabla \bar{\mathcal{L}}(\boldsymbol{\theta}), \boldsymbol{d}^\star \rangle + \bar{\mathcal{R}}(\boldsymbol{\theta} + \boldsymbol{d}^\star) - \bar{\mathcal{R}}(\boldsymbol{\theta}) \leq -(\boldsymbol{d}^\star)^T \nabla^2 \bar{\mathcal{L}}(\boldsymbol{\theta}) \boldsymbol{d}^\star.$$

Proof. Since $\boldsymbol{d}^\star$ is the optimal solution of (20), for any $\alpha < 1$ we have

$$\begin{aligned}\langle \nabla \bar{\mathcal{L}}(\boldsymbol{\theta}) + \boldsymbol{d}^\star \rangle + \frac{1}{2}(\boldsymbol{d}^\star)^T H \boldsymbol{d}^\star + \bar{\mathcal{R}}(\boldsymbol{\theta} + \boldsymbol{d}^s tar) &\leq \langle \nabla \bar{\mathcal{L}}(\boldsymbol{\theta}), \alpha \boldsymbol{d}^\star \rangle + \frac{1}{2}\alpha^2 (\boldsymbol{d}^\star)^T H \boldsymbol{d}^\star + \bar{\mathcal{R}}(\boldsymbol{\theta} + \alpha \boldsymbol{d}^\star) \\ &\leq \alpha \langle \nabla \bar{\mathcal{L}}(\boldsymbol{\theta}), \boldsymbol{d}^\star \rangle + \frac{1}{2}\alpha^2 (\boldsymbol{d}^\star)^T H \boldsymbol{d}^\star + \alpha \bar{\mathcal{R}}(\boldsymbol{\theta} + \boldsymbol{d}) + (1-\alpha)\bar{\mathcal{R}}(\boldsymbol{\theta}),\end{aligned}$$

where we use H to denote the exact Hessian $\nabla^2 \bar{\mathcal{L}}(\boldsymbol{\theta})$ and the second inequality is by the convexity of $\bar{\mathcal{R}}$ since we consider all the atomic norm $\|\cdot\|_{\mathcal{A}}$ to be convex. Therefore

$$(1-\alpha)\left(\langle \nabla \bar{\mathcal{L}}(\boldsymbol{\theta}), \boldsymbol{d}^\star \rangle + \bar{\mathcal{R}}(\boldsymbol{\theta} + \boldsymbol{d}^\star) - \bar{\mathcal{R}}(\boldsymbol{\theta})\right) + \frac{1}{2}(1-\alpha^2)(\boldsymbol{d}^\star)^T H \boldsymbol{d}^\star \leq 0.$$

Dividing both sides by $(1-\alpha)$ we get

$$\delta + \frac{1}{2}(1-\alpha)(\boldsymbol{d}^\star)^T H \boldsymbol{d}^\star \leq 0,$$

therefore $\delta \leq -(\boldsymbol{d}^\star)^T H \boldsymbol{d}^\star$. □

Combining Lemma 3 and (26) we get

$$F(\boldsymbol{\theta} + \boldsymbol{d}^\star) \leq F(\boldsymbol{\theta}) + \frac{\delta}{2} + \frac{1}{6}\eta \|\boldsymbol{d}^\star\|^3.$$

Furthermore, considering $\boldsymbol{\theta}$ in the level set, we can define M to be the largest eigenvalue of Hessians, and thus

$$F(\boldsymbol{\theta} + \boldsymbol{d}^\star) \leq F(\boldsymbol{\theta}) + (\frac{1}{2} - \frac{1}{6}\eta M^2 \|\boldsymbol{d}^\star\|)\delta.$$

Since $\boldsymbol{d} \to 0$ as $\boldsymbol{\theta} \to \boldsymbol{\theta}^\star$, we can find an ϵ-ball around $x^\star$ such that $\frac{1}{2} - \frac{1}{6}\eta M^2 \|\boldsymbol{d}^\star\|) > \sigma$, and $\delta < 0$, thus line search will be satisfied with step size equals to 1.

Based on the above proofs, when $\boldsymbol{\theta}$ is closed enough to $\boldsymbol{\theta}^\star$, $\mathcal{S}_{\text{free}} = \text{span}(\{\boldsymbol{a} \mid \langle \boldsymbol{a}, \boldsymbol{\theta}^\star \rangle\})$ and the step size $\alpha = 1$. Finally we have to explore the structure of dirty model. Since we have k parameter sets $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(k)}$, the Hessian has a block structure presented in (7). Even when $\nabla^2 \mathcal{L}(\boldsymbol{\theta})$ is strongly convex, the rank of the Hessian matrix is at most $O(n)$, while there are totally nk variables or rows. However, our main observation is that when $\nabla^2 \mathcal{L}(\boldsymbol{\theta})$ is positive definite, the Hessian $H \in \mathcal{R}^{pk \times pk}$ has a fixed null space:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}([I, I, \ldots, I][\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(k)}]^T) = \mathcal{L}(E\boldsymbol{\theta}),$$

where $E = [I, I, \ldots, I]$. The null space of H is always the null space of E when $\mathcal{L}$ is strongly convex itself. The following theorem (Theorem 2 in [28] can then be applied:

Lemma 4. *Let $F(\boldsymbol{\theta}) = g(\boldsymbol{\theta}) + h(\boldsymbol{\theta})$ where $g(\boldsymbol{\theta})$ has a constant null space $\mathcal{T}^\perp$ and is strongly convex in the subspace $\mathcal{T}$, and has Lipschitz continuous second order derivative $\nabla^2 g(\boldsymbol{\theta})$. If we apply a proximal Newton method (BCGD-block1 with exact Hessian and step size 1) to minimize $F(\boldsymbol{\theta})$, then*

$$\|\boldsymbol{z}_{t+1} - \boldsymbol{z}^\star\| \leq \frac{L_H}{2m}\|\boldsymbol{z}_t - \boldsymbol{z}^\star\|^2,$$

where $\boldsymbol{z}^\star = proj_{\mathcal{T}}(\boldsymbol{\theta}^\star$, $\boldsymbol{z}_t = proj_{\mathcal{T}}(\boldsymbol{z}_t)$, and L_H is the Lipschitz constant for $\nabla^2 g(\boldsymbol{\theta})$.

In our case, $\text{proj}_{\mathcal{T}}([\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(k)}]^T) = \sum_{i=1}^k \boldsymbol{\theta}^{(i)}$, therefore we have

$$\|\sum_{i=1}^k \boldsymbol{\theta}_{t+1}^{(i)} - \boldsymbol{\theta}^\star\| \leq C\|\sum_{i=1}^k \boldsymbol{\theta}_t^{(i)} - \boldsymbol{\theta}^\star\|^2,$$

therefore $\bar{\boldsymbol{\theta}} = \sum_{i=1}^k \boldsymbol{\theta}^{(i)}$ has an asymptotic convergence rate. □

7.5 Dual of latent GMRF

The problem (4) can be rewritten by

$$\min_{S,L:L\succeq 0,S-L\succ 0} -\log\det(S-L)+\langle S-L,\Sigma\rangle+\max_{Z:\|Z\|_\infty\le\alpha}\operatorname{trace}(ZS)+\max_{P:\|P\|_2\le\beta}\operatorname{trace}(LP),$$

where $\|Z\|_\infty = \max_{i,j}|Z_{ij}|$ and $\|P\|_2 = \sigma_1(P)$ is the induced two norm. We then interchange min and max to get

$$\begin{aligned}\max_{Z,P:\|Z\|_\infty\le\alpha,\|P\|_2\le\beta}\ \min_{L,S,S-L\succ 0,L\succeq 0} &-\log\det(S-L)+\langle S-L,\Sigma\rangle+\operatorname{trace}(ZS)+\operatorname{trace}(LP)\\ &\equiv g(Z,P,L,S). \end{aligned}\tag{27}$$

Assume we do not have the constraint $L\succeq 0$, then the minimizer will satisfy

$$\begin{aligned}\nabla_L g(Z,P,L,S) &= -(S-L)^{-1}+\Sigma+Z=0\\ \nabla_S g(Z,P,L,S) &= -(S-L)^{-1}-\Sigma+P=0.\end{aligned}$$

Therefore we have

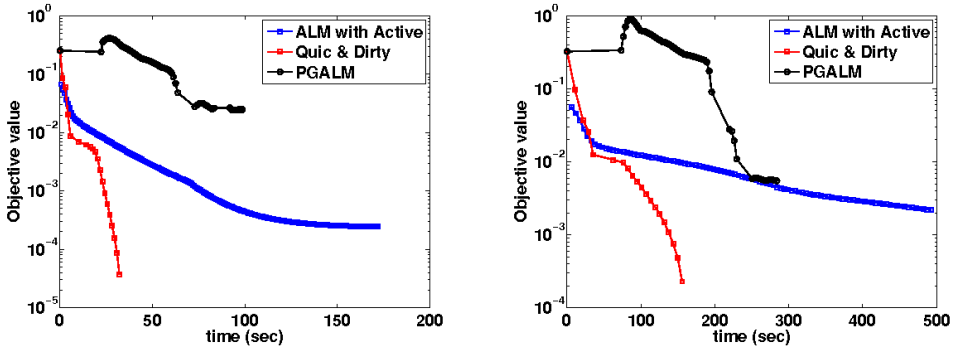
$$Z=-P \ \text{ and } \ S-L=(\Sigma+Z)^{-1}.\tag{28}$$

Combining (28) and (27) we get the dual problem

$$\min_{\Sigma+Z\succeq 0}\log\det(\Sigma+Z)+p \text{ s.t. } \|Z\|_\infty\le\alpha,\|Z\|_2\le\beta.$$

7.6 Alternating Minimization approach for latent GMRF

Another way to solve the latent GMRF problem is to directly applying an alternating minimization scheme to solve (4). The algorithm iteratively fix one of the S, L and update the other. We can still conduct the same active subspace selection technique mentioned in Section 3 in this algorithm. However, this alternating minimization approach can achieve at most linear convergence rate, while our algorithm can achieve super-linear convergence. In this section, we show the detail implementation for this algorithm — ALM(active), and show the comparison results with the Quic & Dirty algorithm (Algorithm 1). The results in Figure 2 shows that our proximal Newton method is faster in terms of the convergence rate.



(a) ER dataset, log scale

(b) Leukemia dataset, log scale

Figure 2: Comparison between QUIC and Alternating Minimization (AM) on gene expression datasets. Note that we implement the active subspace selection approach on both algorithm, so two algorithms have similar speed per iteration. However, we observe that QUIC is much more efficient in terms of final convergence rate.

7.7 Details on the implementation of our multi-task solver

We consider the multi-task learning problem. Assume we have k tasks, each with samples $X^{(r)}\in\mathcal{R}^{d,n_r}$ and labels $\boldsymbol{y}^{(r)}\in\mathcal{R}^{n_r}$. The goal of multi-task learning is to estimate the model $W\in\mathcal{R}^{d\times k}$,

where each column of W, denoted by $\boldsymbol{w}^{(r)}$, is the model for the r-th task. A dirty model has been proposed in [15] to estimate W by $S + B$, where

$$(S, B) = \underset{S,B \in \mathcal{R}^{d\times k}}{\operatorname{argmin}} \|\boldsymbol{y}^{(k)} - X^{(k)}(\boldsymbol{s}^{(k)} + \boldsymbol{b}^{(k)})\|^2 + \lambda_S \|S\|_1 + \lambda_B \|B\|_{1,2}, \tag{29}$$

where $\boldsymbol{s}^{(k)}, \boldsymbol{b}^{(k)}$ are the k-th column of S, B. It was shown in [15] that the combination of sparse and group sparse regularization yields better performance both in theory and in practice.

Instead of considering the squared-loss problem in (29), we further consider optimization problem minimizing the logistic loss:

$$(S, B) = \underset{S,B \in \mathcal{R}^{d\times k}}{\operatorname{argmin}} \sum_{r=1}^{k} \left(\sum_{i=1}^{n_r} \ell_{logistic}(\boldsymbol{y}_i^{(k)}, (\boldsymbol{s}^{(k)} + \boldsymbol{b}^{(k)})^T \boldsymbol{x}_i^{(k)}) \right) + \lambda_S \|S\|_1 + \lambda_B \|B\|_{1,2}, \tag{30}$$

where $\ell_{logistic}(y, a) = \log(1 + e^{-ya})$. This loss function is more suitable for the classification case, as shown in the later experiments.

Let $W = S + B$, and we define $\boldsymbol{w}^{(r)}$ to be the r-th column of W, then the Hessian and gradient of $\mathcal{L}(\cdot)$ can be computed by

$$\nabla_{\boldsymbol{w}^{(r)}} \sum_{i=1}^{n_r} (\sigma(y_i^{(r)} \langle \boldsymbol{w}^{(r)}, \boldsymbol{x}_i^{(r)} \rangle) - 1) y_i^{(r)} \boldsymbol{x}_i^{(r)}, \quad \nabla^2_{\boldsymbol{w}^{(r)}, \boldsymbol{w}^{(r)}} \mathcal{L}(W) = (X^{(r)})^T D^{(r)} X^{(r)},$$

where $\sigma(a) = 1/(1+e^{-a})$ and $D^{(r)}$ is a diagonal matrix with $D_{ii}^{(r)} = \sigma(y_i^{(r)} \langle \boldsymbol{w}^{(r)}, \boldsymbol{x}_i^{(r)} \rangle)$ for all $i = 1, \ldots, n_r$. Note that the Hessian of $\mathcal{L}(W)$ is a block-diagonal matrix, so $\nabla^2_{\boldsymbol{w}^{(r)}, \boldsymbol{w}^{(t)}} \mathcal{L}(W) = 0$. Note also that to form the quadratic approximation at W, we only need to compute $\sigma(y_i^{(r)} \langle \boldsymbol{w}^{(r)}, \boldsymbol{x}_i^{(r)} \rangle)$ for all i, r.

For the sparse-structured parameter component, we select a subset of variables in S to update as in the previous example. For the group-sparse structured component B, we select a subset of "rows" in B to update, following (15). To solve the quadratic approximation subproblem, we again use coordinate descent to minimize with respect to the sparse S component. For the group-sparse component B, we use block coordinate descent, where each time we update variables in one group (one row) using the trust region approach described in [20]. Since $\mathcal{S}_{\text{free}}$ contains only a small subset of blocks, the block coordinate descent can focus on this subset and becomes very efficient.

Algorithm. We first derive the quadratic approximation for the logistic loss function (30). Let $W = S + B$, the gradient for the loss function $\mathcal{L}(W)$ can be written as

$$\nabla_{\boldsymbol{w}^{(r)}} \sum_{i=1}^{n_r} (\sigma(y_i^{(r)} (\boldsymbol{w}^{(r)})^T \boldsymbol{x}_i^{(r)}) - 1) y_i^{(r)} \boldsymbol{x}_i^{(r)},$$

where $\sigma(a) = 1/(1 + e^{-a})$. The Hessian of $\mathcal{L}(W)$ is a block-diagonal matrix, where each $d \times d$ block corresponds to variables in one task, i.e., $\boldsymbol{w}^{(r)}$. Let $H \in \mathcal{R}^{kd \times kd}$ Hessian matrix, each $d \times d$ block can be written as

$$\nabla^2_{\boldsymbol{w}^{(r)}, \boldsymbol{w}^{(r)}} \mathcal{L}(W) = (X^{(r)})^T D^{(r)} X^{(r)},$$

where $D^{(r)}$ is a diagonal matrix with $D_{ii}^{(r)} = \sigma(y_i^{(r)} \langle \boldsymbol{w}^{(r)}, \boldsymbol{x}_i^{(r)} \rangle)$ for all $i = 1, \ldots, n_r$.

Therefore, to form the quadratic approximation of the current solution, the only computation required is to compute $\sigma(y_i^{(r)} \langle \boldsymbol{w}^{(r)}, \boldsymbol{x}_i^{(r)} \rangle)$ for all i, r.

For the lasso part, we select a subset of variables in S to update, according to the subspace selection criterion described 3.2. For the group lasso, we select a subset of "rows" in B to update, as described in 3.2.

For the lasso part, we apply a coordinate descent solver for solving the subproblem. Notice that for the Lasso part each column of S forms a subproblems:

$$\min_{\delta \in \mathbb{R}^d} \frac{1}{2} \delta^T H^{(r)} \delta + \boldsymbol{g}^{(r)} \delta + \|\boldsymbol{s}^{(r)} + \delta\|,$$

where $H^{(r)} = (X^{(r)})^T D^{(r)} X^{(r)}$ and $\boldsymbol{g}^{(r)} = \nabla_{\boldsymbol{s}^{(r)}} \mathcal{L}(W)$. The k subproblems are independent to each other, so we can solve them independently. For each subproblem, we apply the coordinate

descent approach described in [29] to solve it. When update the coordinate δ_i, the key computation is to compute the current gradient $H^{(r)}\delta + \boldsymbol{g}^{(r)}$. Directly computing this is expensive, however, we can maintain $\boldsymbol{p} = D^{(r)}X^{(r)}\delta$ during the updates, and then compute $H^{(r)}\delta = (X^{(r)}_{:,i})^T\boldsymbol{p}$, which only takes $O(\|X^{(r)}_{:,i}\|_0)$ flops.

For solving the group lasso problem, we cannot solve each column independently because the regularization is grouping each row of B. We apply a block coordinate descent method, where each time only one row of B is updated. Let $\delta \in \mathbb{R}^k$ denote the update on the i-th row of B, the subproblem with respect to δ can be written as

$$\frac{1}{2}\sum_{r=1}^{k}\gamma_r(\delta_j)^2 + \boldsymbol{g}^T\delta + \lambda\|\delta + \bar{\boldsymbol{w}}\|, \tag{31}$$

where $\bar{\boldsymbol{w}}$ is the i-th row of W; $\gamma_r = H^{(r)}_{ii}$ and $g_r = \nabla_{W_{ir}}\mathcal{L}(W)$ can be precomputed and will not change during the update.

By taking the gradient of the subproblem (31), we can see that

$$\delta = -\bar{\boldsymbol{w}} + \begin{cases} 0 & \text{if } \|\boldsymbol{g} - \sum_{r=1}^{k}\gamma_r\bar{w}_r^2\| \le \lambda \\ -(\Gamma + \frac{\lambda}{\|\bar{\boldsymbol{w}}+\delta\|}I)^{-1}\boldsymbol{g} & \text{if } \|\boldsymbol{g} - \sum_{r=1}^{k}\gamma_r\bar{w}_r^2\| > \lambda. \end{cases} \tag{32}$$

For the second case, the closed form solution exists when $\Gamma = I$. However, this is no true in general. Instead, we use the iterative trust-region solver proposed in [20] to solve the subproblem, where each iteration of the Newton root finding algorithm only takes $O(k)$ time. Therefore, the computational bottleneck is to compute $\boldsymbol{g}$ in (31). In our case, similar to the Lasso subproblem, we can maintain $\boldsymbol{p}^{(r)} = D^{(r)}X^{(r)}\delta^{(r)}$ for each $r = 1, \ldots, k$ in memory, where $\delta^{(r)}$ is the r-th column of change in W. The gradient can then be computed by $\boldsymbol{g} = H^{(r)}\delta^{(r)} = X^{(r)}\delta^{(r)}$, therefore the time complexity is $O(\bar{n})$ for each coordinate update, where $\bar{n}$ is number of nonzero for each column of $X^{(r)}$.

7.8 Proof of Proposition 1

To prove (a), first we expand the sub-differential

$$\langle \boldsymbol{a}, \partial_{\boldsymbol{\theta}^{(r)}}F(\boldsymbol{\theta})\rangle = \langle \boldsymbol{a}, \partial_{\boldsymbol{\theta}^{(r)}}\mathcal{L}(\bar{\boldsymbol{\theta}}) + \lambda_r\partial_{\boldsymbol{\theta}^{(r)}}\|\boldsymbol{\theta}^{(r)}\|_{\mathcal{A}_r}\rangle = \langle \boldsymbol{a}, G\rangle + \lambda_r\langle \boldsymbol{a}, \rho\rangle \quad \text{for} \quad \rho \in \partial_{\boldsymbol{\theta}^{(r)}}\|\boldsymbol{\theta}^{(r)}\|_{\mathcal{A}_r}$$

and now using the properties of decomposable norms, we calculate

$$\begin{aligned} |\langle \boldsymbol{a}, \rho\rangle| &= |\langle \boldsymbol{a}, \Pi_{\mathcal{T}_r^\perp}\rho\rangle| \\ &\le \|\boldsymbol{a}\|_{\mathcal{A}_r}\|\Pi_{\mathcal{T}_r^\perp}\rho\|^*_{\mathcal{A}_r} \\ &\le 1 \end{aligned}$$

hence $\langle \boldsymbol{a}, G\rangle + \lambda_r\langle \boldsymbol{a}, \rho\rangle \in \langle \boldsymbol{a}, G\rangle - \lambda_r, [\langle \boldsymbol{a}, G\rangle + \lambda_r]$ and the result is shown. In fact, it is not hard to see that *every* element of the set can be written as $\langle \boldsymbol{a}, \rho\rangle$ for some $\rho \in \partial_{\boldsymbol{\theta}^{(r)}}\|\boldsymbol{\theta}^{(r)}\|_{\mathcal{A}_r}$.

To prove (b), note that the optimality condition on σ is that $\sigma^\star$ will satisfy $0 \in \partial_\sigma F(\boldsymbol{\theta} + \sigma^\star\boldsymbol{a})$ and by the chain rule, $\partial_\sigma F(\boldsymbol{\theta} + \sigma\boldsymbol{a}) = \langle \boldsymbol{a}, \partial_{\boldsymbol{\theta}^{(r)}}F(\boldsymbol{\theta} + \sigma\boldsymbol{a})\rangle$. If $\langle \boldsymbol{a}, G\rangle \le \lambda_r$, then by part (a) we see that 0 is in the sub-differential of $\partial_\sigma F(\boldsymbol{\theta})$ and hence $\sigma = 0$ is an optimal point. If F is strongly convex, $\sigma = 0$ is the unique optimal point.

7.9 Proof of Proposition 2

By the definition of proximal operator, in the optimal solution $-G \in \partial_{\boldsymbol{\theta}^{(r)}}\|\text{prox}_{\lambda_r}(G)\|_{\mathcal{A}_r}$. For any $\boldsymbol{a} \in \mathcal{S}^{(r)}_{fixed}$, $\boldsymbol{a} \in \mathcal{T}(\text{prox}^{(r)}_{\lambda_r}(G))^\perp$, thus $|\langle G, \boldsymbol{a}\rangle| < \lambda_r$. Next we consider the projection of gradient to $\mathcal{S}^{(r)}_{fixed}$: let $\rho = \Pi_{S^{(r)}_{fixed}}(G)$, then by the previous statement we know $\|\rho\|_* \le \lambda_r$. Then since $\rho \in \mathcal{T}(\boldsymbol{\theta}^{(r)})^\perp$, we have $\rho \in \lambda_r\partial\|\boldsymbol{\theta}^{(r)}\|_{\mathcal{A}_r}$, therefore constrained to the subspace $\mathcal{S}^{(r)}_{fixed}$, 0 belongs to the sub-gradient, which proves Proposition 2.

7.10 Active Subspace Selection for Group Lasso Regularization

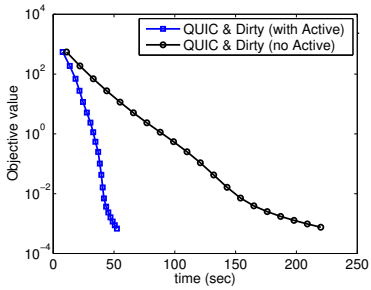


Figure 3: Comparing with/without active subspace selection technique on the RCV1 dataset on the multi-task learning problem with group-lasso regularization and logistic loss. We choose $\lambda = 10^{-3}$ and the final solution only has 1678 nonzero rows, while there are 22283 rows in total.