A Unified Dynamic Approach to Sparse Model Selection

Chendi Huang
School of Mathematical Sciences,
Peking University,
Beijing, China
cdhuang@pku.edu.cn

Yuan Yao
Department of Mathematics,
Hong Kong University of Science and Technology,
HKSAR, China
yuany@ust.hk

Abstract

Sparse model selection is ubiquitous from linear regression to graphical models where regularization paths, as a family of estimators upon the regularization parameter varying, are computed when the regularization parameter is unknown or decided data-adaptively. Traditional computational methods rely on solving a set of optimization problems where the regularization parameters are fixed on a grid that might be inefficient. In this paper, we introduce a simple iterative regularization path, which follows the dynamics of a sparse Mirror Descent algorithm or a generalization of Linearized Bregman Iterations with nonlinear loss. Its performance is competitive to glmnet with a further bias reduction. A path consistency theory is presented that under the Restricted Strong Convexity (RSC) and Irrepresentable Condition (IRR), the path will first evolve in a subspace with no false positives and reach an estimator that is sign-consistent or of minimax optimal $\ell_2$ error rate. Early stopping regularization is required to prevent overfitting. Application examples are given in sparse logistic regression and Ising models for NIPS coauthorship.

1 Introduction

In high dimensional statistics and machine learning, the data $\mathbf{z}$ is often assumed to be generated from a statistical model $P(\alpha^*, \beta^*)$ with a sparse parameter $\beta^*$, and the purpose is to estimate $\beta^*$ typically via the following optimization approach,

$$\min_{\alpha, \beta} \ell(\alpha, \beta; \mathbf{z}) + \lambda P(\beta), \quad (1.1)$$

where $\ell(\alpha, \beta; \mathbf{z})$ is a loss function depending on data $\mathbf{z}$ and parameter $\alpha, \beta$ usually based on likelihood, and $P(\beta)$ is a penalty function. For simplicity, we shall omit the dependence on $\mathbf{z}$ for the loss when it is clear from the context.

Example 1 (Sparse linear regression model). Let $X = (x^{(1)}, \ldots, x^{(n)})^T \in \mathbb{R}^{n \times p}$ be a fixed design matrix, and $y = (y^{(1)}, \ldots, y^{(n)})^T \in \mathbb{R}^n$,

$$y^{(i)} = \alpha^* + \beta^* x^{(i)} + \epsilon^{(i)} \quad (1 \leq i \leq n),$$

with $\epsilon^{(i)}$'s i.i.d. drawn from $N(0, \sigma^2)$, and $\beta^*$ sparse. Let

$$\ell(\alpha, \beta; \mathbf{z}) = \|y - \alpha - X \beta\|_2^2/(2n)$$

be the loss function for data $\mathbf{z} = (x^{(i)}, y^{(i)})_{i=1}^n$ and parameter $(\alpha, \beta)$ (intercept $\alpha$ and linear parameter $\beta$), as well as the Lasso penalty $P(\beta) = \|\beta\|_1$. For model selection consistency, Zhao and Yu (2006) and Wainwright (2009) showed it under Restricted Strong Convexity (RSC) and Irrepresentable Condition (IRR); under a weaker restricted eigenvalue condition, Bickel et al. (2009) established the $\ell_2$-error at minimax optimal rates.

Example 2 (Sparse logistic regression model). Let $x^{(i)} \in \mathbb{R}^p \quad (1 \leq i \leq n)$, and $y^{(i)} \in \{1, -1\}$,

$$\mathbb{P} \left( y^{(i)} = 1 | x^{(i)} \right) = 1 \left( 1 + \exp \left( - (\alpha^* + \beta^* x^{(i)}) \right) \right),$$

with $\beta^*$ sparse. Ravikumar et al. (2010) considered (1.1) with the loss function for data $\mathbf{z} = (y^{(i)}, x^{(i)})_{i=1}^n$ and parameter $(\alpha, \beta)$

$$\ell(\alpha, \beta; \mathbf{z}) = - \frac{1}{n} \sum_{i=1}^n \log \mathbb{P}_{\alpha, \beta} \left( y = y^{(i)} | x^{(i)} \right)$$

$$= \frac{1}{n} \sum_{i=1}^n \log \left( 1 + \exp \left( - (\alpha + \beta^T x^{(i)}) y^{(i)} \right) \right), \quad (1.2)$$
as well as $P(\beta) = \|\beta\|_1$. They also showed its selection/estimation consistency.

**Example 3** (Sparse Ising model). $x^{(i)} (1 \leq i \leq n)$ are drawn from $x \in \{1, -1\}^p$ whose population satisfies

$$
P\left(x = (x_1, \ldots, x_p)^T\right) \propto \exp\left(\frac{1}{2} \sum_{j=1}^{p} \alpha^*_j x_j + \frac{1}{2} \sum_{j < j'} \beta^*_j x_j x_{j'}\right), \quad (1.3)$$

where $\alpha^* \in \mathbb{R}^p$, $\beta^* \in \mathbb{R}^{p \times p}$, and $\beta^*$ is sparse. Ravikumar et al. (2010) studied sparse Ising model (1.3) by the so-called neighborhood-based logistic regression, based on the discussion on sparse logistic models in their paper. Specifically, despite the difficulty to deal with the whole $(\alpha^*, \beta^*)$ by using likelihood-based loss functions of Ising model, they noticed that

$$
P(x_j | x_{-j}) = 1/ (1 + \exp ( - (\alpha^*_j + \beta^*_j x^{(i)}_j) x^{(i)}_j)) .$$

Thus each $j$ corresponds to a sparse logistic regression problem, i.e. Example 2, with $y, x, \alpha, \beta$ replaced by $x_j, x_{-j}, \alpha_j, \beta_{-j,j}$. Thus they learned $(\alpha^*_j, \beta^*_{j,j})$ by $\ell_1$ regularized logistic regression for each $j$, instead of dealing with $(\alpha^*, \beta^*)$ directly. Xue et al. (2012) considered (1.1) with the loss $\ell(\alpha, \beta)$ being the negative composite conditional log-likelihood

$$
\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{p} \log \left( 1 + \exp \left(- \left( \alpha_j + \beta_{-j,j} x_{-j}^{(i)} \right) x_j^{(i)} \right) \right).
$$

$P(\cdot)$ can be $\ell_1$ penalty, SCAD penalty or other positive penalty function defined on $[0, +\infty)$. Alternatively Sohl-Dickstein et al. (2011) proposed an approach of Minimum Probability Flow (MPF) which in the case of Ising model uses the following loss

$$
\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{p} \exp \left(- \frac{1}{2} \left( \alpha_j + \beta_{-j,j} x_{-j}^{(i)} \right) x_j^{(i)} \right).
$$

The minimizer of this function is a reasonable estimator of $(\alpha^*, \beta^*)$. However their work did not treat sparse models in high-dimensional setting. When facing sparse Ising model, one may consider (1.1) with the loss $\ell(\alpha, \beta)$ being the expression in (1.5) and $P(\beta) = \|\beta\|_1$, which is not seen in literature to the best of our knowledge.

**Example 4** (Sparse Gaussian graphical model). $x^{(i)} (1 \leq i \leq n)$ are drawn from a multivariate Gaussian distribution with covariance $\Sigma^* \in \mathbb{R}^{p \times p}$, and the precision matrix $\Omega^* = \Sigma^{-1}$ is assumed to be sparse. Yuan and Lin (2007) and Ravikumar et al. (2008) studied (1.1), with the loss function being the negative scaled log-likelihood, and the penalty being the sum of the absolute values of the off-diagonal entries of the precision matrix.

In general, Negahban et al. (2009) provided a unified framework for analyzing the statistical consistency of the estimators derived by solving (1.1) with a proper choice of $\lambda$. However in practice, since $\lambda$ is unknown, one typically needs to compute the regularization path $\beta_\lambda$ as regularization parameter $\lambda$ varies on a grid, e.g. the lars (Efron et al., 2004) or the coordinate descent in glmnet. Such regularization path algorithms can be inefficient in solving many optimization problems.

In this paper, we look at the following three-line iterative algorithm which, despite its simplicity, leads to a novel unified scheme of regularization paths for all cases above,

$$
\alpha_{k+1} = \alpha_k - \kappa \delta_k \nabla_\alpha \ell(\alpha_k, \beta_k), \quad (1.6a) \\
z_{k+1} = z_k - \delta_k \nabla_\beta \ell(\alpha_k, \beta_k), \quad (1.6b) \\
\beta_{k+1} = \kappa S(z_{k+1}, 1), \quad (1.6c)
$$

where $z_0 = \beta_0 = 0$, $\alpha_0$ can be arbitrary and is naturally set arg min$_\alpha \ell(\alpha, \beta_0)$, step size $\delta_k = \delta$ and $\kappa$ are parameters whose selection to be discussed later, and the shrinkage operator $S(\cdot, 1)$ is defined element-wise as $S(z, 1) = \text{sign}(z) \cdot \max(|z| - 1, 0)$. Such an algorithm is easy for parallel implementation, with linear speed-ups demonstrated in experiment Section 3 below.

To see the regularization paths returned by the iteration, Figure 1 compared it against the glmnet. Such simple iterative regularization paths exhibit competitive or even better performance than the Lasso regularization paths by glmnet in reducing the bias and improving the accuracy (Section 3.2 for more details).

**How does this simple iteration algorithm work?**

There are two equivalent views on algorithm (1.6). First of all, it can be regarded as a mirror descent algorithm (MDA) (Nemirovski and Yudin, 1983; Beck and Teboulle, 2003; Nemirovski, 2012)

$$
(\alpha_{k+1}, \beta_{k+1}) = \arg \min_{z} \{ \langle z, \delta \nabla_{\alpha, \beta} \ell(\alpha_k, \beta_k) \rangle + B_\Phi(\bar{z}, (\alpha_k, \beta_k)) \} \\
:= \text{prox}_{\delta \nabla_{\alpha, \beta} \ell(\alpha_k, \beta_k)}
$$

where $B_\Phi$ is the bregman divergence associated with $\Phi$, i.e. defined by

$$
B_\Phi(u, v) = \Phi(u) - \Phi(v) - \langle \Phi'(v), u - v \rangle. \quad (1.7)
$$

Now set $\Phi(\alpha, \beta) = \|\alpha\|_2^2/(2\kappa) + \|\beta\|_1 + \|\beta\|_2^2/(2\kappa)$ involving a Ridge ($\ell_2$) penalty on $\alpha$ and an elastic net type ($\ell_1$ and $\ell_2$) penalty on $\beta$. Hence $\partial_\alpha \Phi(\alpha, \beta) = \alpha/\kappa$
and \( \partial \Phi(\alpha, \beta) = \rho + \beta / \kappa \) where \( \rho \in \partial ||\beta||_1 \). With this, the optimization in MDA leads to (1.6a) and

\[
\rho_{k+1} = \rho_k + \frac{1}{\kappa} \beta_{k+1} - \delta \nabla \beta \ell(\alpha_k, \beta_k), \quad \rho_k \in \partial ||\beta||_1, \tag{1.8}
\]

which is equivalent to (1.6b). There has been extensive studies on the convergence \( \ell(\alpha_k, \beta_k) - \min_{\beta} \ell(\alpha, \beta) \leq O(k^{-r}) \) \( (r > 0) \), which are however not suitable for statistical estimate above as such convergent solutions lead to overfitting estimators.

It is a restricted gradient flow (differential inclusion) where \( \beta(t) \) has its sparse support controlled by \( \rho(t) \). As \( \kappa \to \infty \), it gives a sequence of estimates by minimizing \( \ell \) with the sign pattern of \( \beta(t) \) restricted on \( \rho(t) \). Thus if an estimator \( \beta(t) \) has the same sign pattern as \( \beta^* \), it must returns the unbiased oracle estimator which is optimal. So it is natural to ask if there is a point on the path \( \beta(t) \) \( (\beta_k) \) which meets the sparsity pattern of true parameter \( \beta^* \). This is the path consistency problem to be addressed in this paper. In Section 2, we shall present a theoretical framework as an answer, and Section 3 gives more applications, including Ising model learning for NIPS coauthorship.

Figure 1: Top: path comparison between \( \{\beta_\alpha\} \ (t = 1 / \lambda) \) by glmnet (left) and \( \{\hat{\beta}_k\} \ (t = k \delta) \) by GLBI (middle), for logistic models with true parameters (right). Bottom: path comparison of \( \{\beta_\alpha\} \ (t = 1 / \lambda) \) by glmnet (left), \( \{\hat{\beta}_k\} \ (t = k \delta) \) by GLBI + composite loss (middle left), and \( \{\hat{\beta}_k\} \ (t = k \delta) \) by GLBI + MPF loss (middle right), for Ising models with true parameters (right). In both cases, glmnet selects biased estimates while GLBI finds more accurate ones.

An alternative dynamic view may lead to a deeper understanding of the regularization path. In fact for Example 1, (1.6) reduces to the Linearized Bregman Iteration (LBI) proposed by Yin et al. (2008) and analyzed by Osher et al. (2016) via its limit differential inclusions. It shows that equipped with the standard conditions as Lasso, an early stopping rule can find a point on the regularization path of (1.6) with the same sign pattern as true parameter (sign-consistency) and gives the unbiased oracle estimate, hence better than Lasso or any convex regularized estimates which are always biased. This can be generalized to our setting where (1.8) is a discretization of the following dynamics

\[
\dot{\alpha}(t) = - \nabla \alpha \ell(\alpha(t), \beta(t)), \quad \dot{\beta}(t) = - \nabla \beta \ell(\alpha(t), \beta(t)), \quad \rho(t) \in \partial ||\beta||_1. \tag{1.9}
\]

2 Path Consistency of GLBI

Let \( \theta^* = (\alpha^*, \beta^* T)^T \) denotes the true parameter, with sparse \( \beta^* \). Define \( S := \text{supp}(\beta^*) \ (s := |S| < p) \) as the index set corresponding to nonzero entries of \( \beta \), and \( S^c \) be its complement. Let \( S_\alpha = (\alpha, S) \), and \( S_\alpha = S \) when \( \alpha \) drops. Let the oracle estimator be

\[
\theta^o = (\alpha^o, \beta^o T)^T \in \arg \min_{\alpha, \beta} \ell(\alpha, \beta), \tag{2.1}
\]

which is an optimal estimate of \( \theta^* \). GLBI starts within the oracle subspace \( \{\theta = (\alpha, \beta^T) : \beta_{Sc} = 0\} \), and we are going to prove that under an Irrepresentable Condition (IRR) the dynamics will evolve in the oracle subspace with high probability before the stopping time \( \bar{k} \), approaching the oracle estimator exponentially fast due to the Restricted Strong Convexity (RSC). Thus if all the true parameters are large enough, then we can identify their sign pattern correctly; otherwise, such a stopping time still finds an estimator (possibly with false positives) at minimax optimal \( \ell_2 \) error rate. Furthermore, if the algorithm continues beyond the stopping time, it might escape the oracle subspace.
and eventually reach overfitted estimates. Such a picture is illustrated in Figure 2.

Figure 2: An illustration of global dynamics of the algorithm in this paper.

Hence, it is helpful to define the following oracle dynamics:

\begin{align}
\alpha_k' &= \alpha_k - \kappa \delta \nabla \ell (\alpha_k', \beta_k' ) , \quad (2.2a) \\
z_k' &= z_k - \delta \nabla S (\alpha_k', \beta_k' ) , \quad (2.2b) \\
\beta_k' &= \kappa S (z_k', 1) , \quad (2.2c)
\end{align}

with \( z_k' = \beta_k' \delta / \kappa \). Let \( \theta_k' := (\alpha_k', \beta_k')^T \).

2.1 Basic Assumptions

Now we are ready to state the general assumptions that can be reduced to existing ones. We write

\[ \ell(\theta) := \ell(\alpha, \beta), \]
\[ \tilde{H}(\alpha, \beta) := \int_0^1 \nabla^2 \ell (\theta^* + \mu (\theta - \theta^*)) \, d\mu, \]
\[ \tilde{H}(\alpha, \beta) := \int_0^1 \nabla^2 \ell (\theta^* + \mu (\theta - \theta^*)) \, d\mu. \]

Assumption 1 (Restricted Strong Convexity (RSC)). There exist \( \lambda, \Lambda > 0 \), such that for any \( k \geq 0 \), and for any \( \theta \on the line segment between \( \theta_k' \) and \( \theta^o \), or on the line segment between \( \theta^* \) and \( \theta^o \),

\[ \lambda I \preceq \nabla^2 \ell (\theta) \preceq \Lambda I, \]

Assumption 2 (Irrepresentable Condition (IRR)). There exist \( \eta \in (0, 1] \) and \( C > 0 \) such that

\[ \sup_{k \geq 1} \left| \sum_{k=0}^{K-1} \nabla \tilde{H}_k \right| \leq C, \]

where

\[ \nabla \tilde{H}_k := \tilde{H}_k (\theta_k') \cdot \tilde{H}_k (\theta_k'). \]

Remark 1. For sparse linear regression problem (Example 1) with no intercept \( (\alpha \) drops), Assumption 1 reduces to \( \lambda I \preceq X_S'X_S \preceq \Lambda I \). The lower bound is exactly the RSC proposed in linear problems. Although the upper bound is not needed in linear problems, it arises in the analysis for logistic problem by Ravikumar et al. (2010) (see (A1) in Section 3.1 in their paper). Besides, \( \text{irr}_k \) is constant and Assumption 2 reduces to

\[ \sup_{k \geq 1} \left| X_S'X_S (X_S'X_S)^{-1} z_k' \right| _\infty < 1 - \eta / 2, \]

which is true with high probability, as long as the classical Irrepresentable Condition (Zhao and Yu, 2006) \( \| X_S'X_S (X_S'X_S)^{-1} \| _\infty < 1 - \eta \) holds along with \( C \geq 1 \) and \( \kappa \) is large, since by (C.6),

\[ \left| z_k' \right| _\infty \leq \left| z_k - S (z_k', 1) \right| _\infty + \left| S (z_k', 1) \right| _\infty \leq 1 + \| \beta_k' \| _\infty / \kappa \]

\[ \leq 1 + \left( \| \beta_k' \| _\infty / \kappa \right) \]

\[ \leq 1 + \left( \frac{\sqrt{\lambda/\lambda + 1}}{\kappa} \| \beta_k' \| _\infty / \kappa \right) \]

\[ < (1 - \eta / 2) / (1 - \eta). \]

Remark 2. For sparse logistic regression problem (Example 2), we have the following proposition stating that Assumption 1 and 2 hold with high probability under some natural setting, along with condition (2.3). See its proof in Appendix D. A slightly weaker condition compared to (2.3b), and a same version of (2.3c), can be found in Ravikumar et al. (2010), where \( x^{(i)} \) are discrete.

Proposition 1. In Example 2, we suppose \( x^{(i)} \)'s are i.i.d. drawn from some \( X \sim N(0, \Sigma) \), where \( \Sigma_{i,j} \leq 0 \) \( (1 \leq j \leq p) \). Then there exist constants \( C_0, C_1, C_2 > 0 \), such that Assumption 1 and 2 hold with probability not less than \( 1 - C_0 / p \), as long as \( \kappa \) is sufficiently large and

\[ \| \Sigma_{S',S} \Sigma_{S,S}^{-1} \| _\infty \leq 1 - \eta, \quad (2.3a) \]

\[ n/(\log n)^3 \geq C_1 s^4 \log p, \quad (2.3b) \]

\[ \beta^{\ast}_{\min} := \min_{j \in S} | \beta_{\ast}^j | \geq C_2 \sqrt{s \log p}/n. \quad (2.3c) \]

2.2 Path Consistency Theorem

Theorem 1 (Consistency of GLBI). Under Assumption 1 and 2, suppose \( \kappa \geq 2 \| \theta_{\ast}^S \| _2 \), and \( \bar{k} \in \mathbb{N} \) such that

\[ \frac{k - 1}{2C + 1} \delta < \frac{\eta}{2C + 1} \cdot \frac{1}{\| \nabla \ell (\alpha_{\ast}, \beta_{\ast}) \| _\infty} \leq \bar{k} \delta. \quad (2.4) \]
Define $\mathcal{X}$ as in (C.3). We have the following properties.

No-false-positive: For all $0 \leq k \leq \bar{k}$, the solution path of GLBI has no false-positive, i.e. $\beta_{k,s'} = 0$.

Sign consistency: If $\bar{k} \geq 5$ and

$$\beta_{\min}^* := \min_{j \in S} |\beta_j^*| \geq \max \left( 2 \| \beta_S^* - \beta_S^o \|_\infty, \frac{(8 \log s + 18)(C + 1)}{\lambda \eta (1 - 4/k)} \| \nabla \ell (\alpha^*, \beta^*) \|_\infty \right),$$

(2.5)

then $\text{sign}(\beta_k) = \text{sign}(\beta^*)$.

$\ell_2$ consistency: The $\ell_2$ error

$$\left\| \frac{\alpha_k - \alpha^o}{\beta_k - \beta^o} \right\|_2 \lesssim \frac{20(C + 1)\sqrt{s}}{\lambda \eta} \| \nabla \ell (\alpha^*, \beta^*) \|_\infty.$$

The proof of Theorem 1 is collected in Appendix C, which largely follows the analysis of differential inclusion (1.6), given in Appendix B, as its discretization.

Remark 3. For sparse linear regression problem (Example 1), with high probability we have

$$\| \nabla \ell (\alpha^*, \beta^*) \|_\infty \lesssim \sqrt{\frac{\log p}{n}}, \| \beta_S^* - \beta_S^o \|_\infty \lesssim \sqrt{\frac{\log s}{n}}.$$

Hence pick $\bar{k} \delta \sim \sqrt{n/\log p}$ satisfying (2.4), by Theorem 1 the sign consistency is guaranteed at $\bar{k}$ if

$$\beta_{\min}^* \gtrsim (\log s) \sqrt{\log p}/n.$$

The $\ell_2$ error bound reaches the minimax optimal rate:

$$\left\| \frac{\alpha_k - \alpha^o}{\beta_k - \beta^o} \right\|_2 \lesssim \sqrt{\frac{s \log p}{n}}.$$

Remark 4. For sparse logistic regression problem (Example 2), with high probability we have

$$\| \nabla \ell (\alpha^*, \beta^*) \|_\infty \lesssim \sqrt{\frac{\log p}{n}}, \| \beta_S^* - \beta_S^o \|_\infty \lesssim \sqrt{s} \| \nabla \ell (\alpha^*, \beta^*) \|_\infty \lesssim \sqrt{s (\log p)/n}.$$

Hence the sign consistency is guaranteed at some $\bar{k} \sim \sqrt{n/\log p}$ if

$$\beta_{\min}^* \gtrsim \sqrt{s (\log p)/n}$$

(meeting Condition (19) in Ravikumar et al. (2010)). The $\ell_2$ error rate $\lesssim \sqrt{(s \log p)/n}$ is minimax optimal.

3 Experiments

As for the setting of algorithm parameters: $\kappa$ should be large, and then $\delta \sim 1/(\kappa \Lambda)$ is automatically calculated based on $\kappa$ (as long as $\kappa \delta \Lambda < 2$, such that $\lambda^*$ is positive in (C.3)). In practice, a small $\delta$ can prevent the iterations from oscillations.

3.1 Efficiency of Parallel Computing

Osher et al. (2016) has elaborated that LBI can easily be implemented in parallel and distributed manners, and applied on very large-scale datasets. Likewise, GLBI can be parallelized in many usual applications. We now take the logistic model Example 2 as an example to explain the details. The iteration (1.6) (generally taking $\delta_k = \delta$) can be written as

$$\begin{align*}
\alpha_{k+1} &= \alpha_k - \kappa \delta f (\alpha_k, X \beta_k), \\
z_{k+1} &= z_k - \delta X^T g (\alpha_k, X \beta_k), \\
\beta_{k+1} &= \kappa S (z_{k+1}, 1),
\end{align*}$$

(3.1)

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^{n+1} \to \mathbb{R}^n$ such that

$$\begin{align*}
g (\alpha, w)_i &= -\frac{1}{n} \frac{1}{1 + \exp (- (\alpha + w_i) y^{(i)})} y^{(i)} \in \mathbb{R} \\
&= -\frac{1}{n} \sum_{i=1}^n \frac{1}{1 + \exp (- (\alpha + w_i) y^{(i)})} y^{(i)}.
\end{align*}$$

Suppose $X = [X_1, X_2, \ldots, X_L] \in \mathbb{R}^{n \times p}$, where $X_i$’s are submatrices stored in a distributed manner on a set of networked workstations. The sizes of $X_i$’s are flexible and can be chosen for good load balancing. Let each workstation $l$ hold data $y$ and $X_l$, and variables $z_{k,l}$ and $X_l \beta_{k,l}$ which are parts of $z_k$ and summands of $w_k := X \beta_k$, respectively. The iteration (3.1) is carried out as

$$\begin{align*}
\alpha_{k+1} &= \alpha_k - \kappa \delta f (\alpha_k, w_k), \\
z_{k+1,l} &= z_{k,l} - \delta X_l^T g (\alpha_k, w_k), \\
w_{k+1,l} &= \kappa X_l S (z_{k+1,l}, 1) \\
w_{k+1} &= \sum_{l=1}^L w_{k+1,l} \quad \text{(all-reduce summation)},
\end{align*}$$

where the all-reduce summation step collects inputs from and then returns the sum to all the $L$ workstations. It is the sum of $L$ $n$-dimensional vectors. Therefore, the communication cost is independent of $p$ no matter how the all-reduce step is implemented. It is important to note that the algorithm is not changed at all. particularly, increasing $L$, does not increase the number of iterations. So the parallel implementation is truly scalable.

If $t_L$ denotes the time cost of a single GLBI run with $L$ workstations under the same dataset and the same algorithmic settings, it is expected that $t_L \sim 1/L$. 
Here we show this by an example. Construct a logistic model in Section 3.2 with \(M = 1\), \(r = 0.25\), and three settings for \((p, s, n): (I) (2000, 200, 6000), (II) (5000, 500, 15000), (III) (10000, 1000, 30000)\). For each setting, we run our parallelized version of GLBI algorithm written in C++, with \(k_{\text{max}} = 1000k_0\), where \(k_0\) is the maximal \(k\) such that \(\beta_1 = \cdots = \beta_k = 0\), and the path is early stopped at the \(k_{\text{max}}\)-th iteration. The recorded \(t_L\)'s are shown in Figure 3. The left panel shows \(t_L\) (in seconds) while the right panel shows \(t_1/t_L\), for \(L = 1, \ldots, 8\). We see truly \(t_L \sim 1/L\), which is expected in our parallel and distributed treatment. When \(L\) is large, our package can deal with very large scale problems.

Figure 3: Time cost illustration for logistic model with three settings (black for Setting (I), blue for Setting (II) and red for Setting (III)). In each setting, the left panel shows \(t_L\) while the right panel shows \(t_1/t_L\), for \(L = 1, \ldots, 8\).

### 3.2 Application: Logistic Model

We do rep = 20 independent experiments, in which we construct a logistic model (Example 2), and then compare GLBI with other methods. Specifically, suppose that \(\beta^*\) has a support set \(S = \{1, \ldots, s\}\) without loss of generality. \(\alpha^*, \beta^*_j \ (j \in S)\) are independent, each has a uniform distribution on \([-2M, -M] \cup [M, 2M]\). Each row of \(X \in \mathbb{R}^{n \times p}\) is i.i.d. sampled from \(N(0, \Sigma)\), where \(\Sigma\) is a Toeplitz matrix satisfying \(\Sigma_{j,k} = \rho^{|j-k|}\). When \(X\) and \((\alpha^*, \beta^*)\) are determined, we generate \(y \in \mathbb{R}^n\) as in Example 2.

After getting the sample \((X, y)\), consider GLBI (1.6) and \(\ell_1\) optimization (1.1), both with logistic loss (1.2). For GLBI, set \(\kappa = 10\). For (1.1), apply a grid search for differently penalized problems, for which we use \texttt{glmnet} – a popular package available in Matlab/R that can be applied on \(\ell_1\) regularization for sparse logistic regression models.

For each algorithm, we use \(K\)-fold (\(K = 5\)) cross validation (CV) to pick up an estimator from the path calculated based on the smallest CV estimate of prediction error. Specifically, we split the data into \(K\) roughly equal-sized parts. For a certain position on paths (\(t\) for GLBI, or \(\lambda\) for \texttt{glmnet}) and \(k \in \{1, \ldots, K\}\), we obtain a corresponding estimator based on the data with the \(k\)-th part removed, use the estimator to build a classifier, and get the mis-classification error on the \(k\)-th part. Averaging the value for \(k \in \{1, \ldots, K\}\), we obtain the CV estimate of prediction error, for the obtained estimator corresponding to a certain position on paths. Among all positions, we pick up the estimator producing the smallest CV estimate of prediction error. Besides, we calculate \(\text{AUC (Area Under Curve)}\), for evaluating the path performance of learning sparsity patterns without choosing a best estimator.

Results for \(p = 80, s = 20, M = 1\) are summarized in Table 1. We see that in terms of CV estimate of prediction error, GLBI is generally better than \texttt{glmnet}. Besides, GLBI is competitive with \(\ell_1\) regularization method in variable selection, in terms of AUC. Similar observations for more settings are listed in Table 4, 5 and 6 in Appendix G. Apart from these tables, we can also see the outperformance of CV estimate of prediction error Figure 5 in Appendix G, while in that figure we can see that GLBI further reduces bias, as well as provides us a relatively good estimator with small prediction error if a proper early stopping is equipped.

<table>
<thead>
<tr>
<th>(r)</th>
<th>(n)</th>
<th>GLBI</th>
<th>\texttt{glmnet}</th>
<th>GLBI</th>
<th>\texttt{glmnet}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>400</td>
<td>.9902</td>
<td>.9906</td>
<td>.1221</td>
<td>.1355</td>
</tr>
<tr>
<td>800</td>
<td>.9991</td>
<td>.9990</td>
<td>.1082</td>
<td>.1132</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>400</td>
<td>.9690</td>
<td>.9681</td>
<td>.1321</td>
<td>.1379</td>
</tr>
<tr>
<td>800</td>
<td>.9925</td>
<td>.9921</td>
<td>.1139</td>
<td>.1197</td>
<td></td>
</tr>
</tbody>
</table>

### 3.3 Application: Ising Model with 4-Nearest-Neighbor Grid

We do rep = 20 independent experiments, in which we construct an ising model (Example 3), and then compare GLBI with other methods. Specifically, construct an \(N \times N\) 4-nearest neighbor grid (with periodic boundary conditions) to be graph \(G\), with node set \(V\) and edge set \(E\). The distribution of a random vector \(x\) is given by (1.3) \((p = N^2)\), where \(\alpha^*_j\)'s and
Table 2: Comparisons of GLBI1 (GLBI + composite), GLBI2 (GLBI + MPF), and glmnet, for Ising models with \( p = 36 \). For each algorithm, we run \( rep = 20 \) independent experiments.

<table>
<thead>
<tr>
<th>( T )</th>
<th>( n )</th>
<th>AUC</th>
<th>GLBI1</th>
<th>GLBI2</th>
<th>glmnet</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.25</td>
<td>500</td>
<td></td>
<td>.9754</td>
<td>.9867</td>
<td>.9774</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(.0277)</td>
<td>(.0128)</td>
<td>(.0265)</td>
</tr>
<tr>
<td>750</td>
<td></td>
<td>.9868</td>
<td>.9919</td>
<td>.9891</td>
<td>.9914</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(.0137)</td>
<td>(.0082)</td>
<td>(.0134)</td>
</tr>
<tr>
<td>1.5</td>
<td>500</td>
<td>.9915</td>
<td>.9963</td>
<td>.9929</td>
<td>.9963</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(.0110)</td>
<td>(.0033)</td>
<td>(.0104)</td>
</tr>
<tr>
<td>750</td>
<td></td>
<td>.9963</td>
<td>.9980</td>
<td>.9975</td>
<td>.9980</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(.0041)</td>
<td>(.0029)</td>
<td>(.0042)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( T )</th>
<th>( n )</th>
<th>2nd order MDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.25</td>
<td>500</td>
<td>GLBI1</td>
</tr>
<tr>
<td>750</td>
<td></td>
<td>.9762</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.0079)</td>
</tr>
<tr>
<td>1.5</td>
<td>500</td>
<td>.9840</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.0053)</td>
</tr>
<tr>
<td>750</td>
<td></td>
<td>.9655</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.0087)</td>
</tr>
</tbody>
</table>

\( \beta_{j,j'}^* \)'s \((j,j') \in E\) are i.i.d. and each has a uniform distribution on \([-2/T, -1/T] \cup [1/T, 2/T]\). Let \( X \in \mathbb{R}^{n \times p} \) represents \( n \) samples drawn from the distribution of \( x \) via Gibbs sampling.

After getting the sample \( X \), consider GLBI1 (GLBI with composite loss (1.4)), GLBI2 (GLBI with MPF loss (1.5)) and \( \ell_1 \) optimization (1.1) with logistic loss (see Example 3 for neighborhood-based logistic regression applied on Ising models, or see Ravikumar et al. (2010)). For GLBI1 and GLBI2, set \( \kappa = 10 \). For (1.1), apply a grid search for differently penalized problems; we still use glmnet.

For each algorithm, we calculate the AUC (Area Under Curve), popular for evaluating the path performance of learning sparsity patterns. Besides, we apply K-fold \((K = 5)\) cross validation (CV) to pick up an estimator from the path, with the largest CV estimate of 2nd order marginal distribution correlation (2nd order MDC) in the same way as the CV process done in Section 3.2, here the 2nd order MDC, defined in the next paragraph, is calculated based on two samples of the same size: the \( k \)-th part original data, and the newly sampled Ising model data (based on learned parameters) with the same size of the \( k \)-th part.

For any sample matrix \( X' \in \{1, -1\}^{n' \times p} \), we construct \( d_2(X') \), the 2nd marginal empirical distribution matrix of \( X' \), defined as follows. \( d_2(X') = \langle d_2(X'_1) \rangle_{p \times p} \in \mathbb{R}^{p \times 2p} \), where

\[
d_2(X'_{j_1, j_2}) = \frac{1}{n'} \sum_{i=1}^{n'} \begin{pmatrix}
1 & (x^{(i)}_{j_1} x^{(i)}_{j_2}) = (1,1) \\
1 & (x^{(i)}_{j_1} x^{(i)}_{j_2}) = (1,-1) \\
1 & (x^{(i)}_{j_1} x^{(i)}_{j_2}) = (-1,1) \\
1 & (x^{(i)}_{j_1} x^{(i)}_{j_2}) = (-1,-1)
\end{pmatrix}.
\]

(3.2)

For any sample matrices \( X_1, X_2 \) with the same sample size, we call the correlation between \( \text{vec}(d_2(X_1)) \) and \( \text{vec}(d_2(X_2)) \) the 2nd order marginal distribution correlation (2nd order MDC). This value is expected to be large as well as close to 1 if \( X_1, X_2 \) come from the same model.

Results for \( p = N^2 = 36 \) are summarized in Table 2. GLBI with composite/MPF loss are competitive with or better than glmnet. Similar observations are listed in Table 7 in Appendix G.

3.4 Application: Coauthorship Network in NIPS

Consider the information of papers and authors in Advances in Neural Information Processing Systems (NIPS) 1987–2016, collected from https://www.kaggle.com/benhamner/nips-papers. After preprocessing (e.g. author disambiguation), for simplicity, we restrict our analysis on the most productive \( p = 30 \) authors (Table 3) in the largest connected component of a coauthorship network that two authors are linked if they coauthored at least 2 papers (Coauthorship (2)). The first panel of Figure 4 shows this coauthorship network with edge width in proportion to the number of coauthored papers. There are \( n = 1,028 \) papers authored by at least one of these persons.

Let the \( j \)-th entry of \( x^{(i)} \in \mathbb{R}^p \) be 1 if the \( j \)-th person is involved in the authors of the \( i \)-th paper, and \(-1\) otherwise. Now we fit the data \( x^{(i)} \) \((1 \leq i \leq n)\) by a sparse Ising model (1.3) with parameter \((\hat{\alpha}, \hat{\beta})\). Note that \( \hat{\beta}_{j,j'} = 0 \) indicates that \( j \) and \( j' \) are conditional independent on coauthorship, given all the other authors; \( \hat{\beta}_{j,j'} > 0 \) implies that \( j \) and \( j' \) coauthored more often than their averages, while \( \hat{\beta}_{j,j'} < 0 \) says the opposite.

The right three panels in Figure 4 compares some sparse Ising models chosen from three regularization paths at a similar sparsity level (the percentage of learned edges over the complete graph, here about 12% ~ 14%): GLBI1 (GLBI with composite loss), GLBI2 (GLBI with MPF loss), and \( \ell_1 \) regularization
A Unified Dynamic Approach to Sparse Model Selection

Table 3: Most productive $p = 30$ authors in the largest connected component of Coauthorship (2).

<table>
<thead>
<tr>
<th></th>
<th>First Name</th>
<th>Last Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Michael Jordan 16</td>
<td>Inderjit Dhillon</td>
</tr>
<tr>
<td>02</td>
<td>Bernhard Schölkopf 17</td>
<td>Ruslan Salakhutdinov</td>
</tr>
<tr>
<td>03</td>
<td>Geoffrey Hinton 18</td>
<td>Tong Zhang</td>
</tr>
<tr>
<td>04</td>
<td>Yoshua Bengio 19</td>
<td>Thomas Griffiths</td>
</tr>
<tr>
<td>05</td>
<td>Zoubin Ghahramani 20</td>
<td>David Blei</td>
</tr>
<tr>
<td>06</td>
<td>Terrence Sejnowski 21</td>
<td>Rémi Munos</td>
</tr>
<tr>
<td>07</td>
<td>Peter Dayan 22</td>
<td>Joshua Tenenbaum</td>
</tr>
<tr>
<td>08</td>
<td>Alex Smola 23</td>
<td>Lawrence Carin</td>
</tr>
<tr>
<td>09</td>
<td>Andrew Ng 24</td>
<td>Eric Xing</td>
</tr>
<tr>
<td>10</td>
<td>Francis Bach 25</td>
<td>Richard Zemel</td>
</tr>
<tr>
<td>11</td>
<td>Michael Mozer 26</td>
<td>Martin Wainwright</td>
</tr>
<tr>
<td>12</td>
<td>Pradeep Ravikumar 27</td>
<td>Yoram Singer</td>
</tr>
<tr>
<td>13</td>
<td>Tommi Jaakkola 28</td>
<td>Han Liu</td>
</tr>
<tr>
<td>14</td>
<td>Klaus-Robert Müller 29</td>
<td>Satinder Singh</td>
</tr>
<tr>
<td>15</td>
<td>Yee Teh 30</td>
<td>Christopher Williams</td>
</tr>
</tbody>
</table>

(glmnet), respectively. For more learned graphs from these paths, see Figure 6 in Appendix G. In GLBI1 and GLBI2, set $\kappa = 10$.

We see that all the learned graphs capture some important coauthorships, such as Pradeep Ravikumar (12) and Inderjit Dhillon (16) in a thick green edge in all the three learned graphs, indicating that they collaborated more often than separately for NIPS. Besides, the most productive author Michael Jordan (01) has coauthored with a lot of other people, but is somewhat unlikely to coauthor with several other productive scholars like Yoshua Bengio (04), Terrence Sejnowski (06), etc., indicating by the red edges between Jordan and those people. Further note the edge widths in the second and third graphs are significantly larger than those in the fourth graph, implying that at a similar sparsity level, GLBI tends to provide an estimator with larger absolute values of entries than that by glmnet. That is because under similar sparsity patterns, GLBI may give low-biased estimators.

Acknowledgements

This work is supported in part by National Basic Research Program of China (Nos. 2015CB85600, 2012CB825501), NNSF of China (Nos. 61370004, 11421110001), HKRGC grant 16303817, as well as grants from Tencent AI Lab, Si Family Foundation, Baidu BDI, and Microsoft Research-Asia.
References


Sohl-Dickstein, Jascha, Peter Battaglino, and Michael DeWeese (2011). “Minimum Probability Flow Learning”. In: ed. by Lise Getoor and Tobias Schef-