FAST PROXIMAL ALGORITHMS FOR SELF-CONCORDANT FUNCTION MINIMIZATION WITH APPLICATION TO SPARSE GRAPH SELECTION

Anastasios Kyrillidis and Volkan Cevher
École Politechnique Fédérale de Lausanne
{anastasios.kyrillidis,volkan.cevher}@epfl.ch

ABSTRACT

The convex $\ell_1$-regularized log det divergence criterion has been shown to produce theoretically consistent graph learning. However, this objective function is challenging since the $\ell_1$-regularization is nonsmooth, the log det objective is not globally Lipschitz gradient function, and the problem is high-dimensional. Using the self-concordant property of the objective, we propose a new adaptive step size selection and present the (F)PS ((F)ast Proximal algorithms for Self-concordant functions) algorithmic framework which has linear convergence and exhibits superior empirical results as compared to state-of-the-art first order methods.

Index Terms— Sparse inverse covariance estimation, self-concordance, step size selection

1. INTRODUCTION

Problem setup: Let $X = \{X_1, X_2, \ldots, X_n\}$ be a set of variables with joint Gaussian distribution $f(X_1, X_2, \ldots, X_n) \sim N(\mu, \Sigma)$ where $\mu \in \mathbb{R}^n$ is assumed known and $\Sigma \in \mathbb{R}^{n \times n}$, $\Sigma > 0$ denotes the unknown covariance matrix. In this setting, assume we only have access to the underlying model through a set of independent and identically distributed (iid) samples $\{x_j\}_{j=1}^p$ such that $x_j \sim N(\mu, \Sigma)$, $\forall i$. Given $\{x_j\}_{j=1}^p$, we are interested in inferring any conditional dependencies among $X$ by estimating $\Sigma^{-1}$. A non-robust estimate of $\Sigma^{-1}$ is through the sample covariance $\hat{\Sigma} = \frac{1}{p} \sum_{j=1}^p (x_j - \hat{\mu})(x_j - \hat{\mu})^T$ where $\hat{\mu} = \frac{1}{p} \sum_{j=1}^p x_j$. Unfortunately, in many cases, we cannot afford to acquire adequate samples for accurate $\Sigma^{-1}$ estimation via $\hat{\Sigma}$; for $p \ll n$, $\hat{\Sigma}$ is rank-deficient and the use of sophisticated estimation procedures is imperative.

Graphical models interpretation: In undirected graphical models, each variable $X_i$ corresponds to a node in a Gaussian Markov random field (GMRF). Moreover, let $E = \{(i, j) : X_i \notin X_j \mid X_k \text{ is observed } \forall k \neq i, j\}$ be the set of edges in the graph. Under this setting, we desire to infer the graph structure given a set of observations. Due to the Gaussianity assumption, $\Sigma^{-1} \neq 0 \iff (i, j) \notin E$.

Optimization criteria: [1] shows that the maximum likelihood estimation $(\Sigma^*)^{-1} = \arg \max_{\Sigma \succ 0} \prod_{i=1}^p f(x_i)$ is equivalent to:

$$\Theta^* = \arg \max_{\Theta \succ 0} \left\{ -\log \det(\Theta) + \text{tr}(\Theta \hat{\Sigma}) \right\}$$

where $\Theta^* = (\Sigma^*)^{-1}$. Based on (1), developments in random matrix theory [2] divulge the poor performance of $\Theta^*$ without regularization: the solution to (1) is usually fully dense and no inference about the graph structure is possible. Moreover, when $p \ll n$, the absence of a regularization term leads to non-robust estimates of $\Sigma^{-1}$.

In practice though, parsimonious solutions that adequately explain the data, increase the interpretability of the results even if they lead to worse-valued less objective values. Using $\ell_1$-norm to regularize the objective, (1) can be well-approximated by:

$$\Theta^* = \arg \min_{\Theta \succ 0} \left\{ F(\Theta) := f(\Theta) + g(\Theta) \right\}$$

where $f(\Theta) := -\log \det(\Theta) + \text{tr}(\Sigma \Theta)$ and $g(\Theta) := \rho |\text{vec}(\Theta)|$; with $\rho > 0$ that defines the sparsity of the graph selection.

Challenges: Within this context, the main challenges in (2) are:

- High-dimensional problems have become the norm in data analysis; thus, time- and memory-efficient schemes are crucial.
- Apart from its computational challenge, (2) is a non-trivial convex problem: $f(\Theta)$ is a strictly convex but not globally Lipschitz-continuous gradient function; moreover, $g(\Theta)$ is a nonsmooth regularizer. Even in simple gradient descent schemes, Lipschitz-based optimal step size calculation becomes infeasible and heuristics lead to slowly convergent, state-of-the-art algorithms [3]. Moreover, (2) is constrained over the set of positive-definite matrices and the choice of regularization parameter $\rho$ is crucial [4].

Prior work: Being a special case of semidefinite programming, (2) can be solved using off-the-shelf interior point approaches [5, 6]. Though, the resulting per iteration complexity for existing interior point methods is $O(n^3)$ [7]. This has led to the development of multifarious works, which can be roughly categorized into five camps: (i) first-order gradient methods [7, 8, 9], (ii) second order (Newton-based) gradient methods [10, 11], (iii) interior point-based schemes [12], (iv) Lagrangian [13, 3] and (iii) greedy approaches [14].

While many of the first-order approaches are slowly convergent and require numerous parameters to be set a priori (reducing their universality), recent developments on second-order methods have resulted in very fast solvers. Though, to achieve this fast performance, these approaches “sacrifice” their universality for faster implementation: one can envision complicated examples (e.g., non-modular regularization) where second-order approaches fail to use their “arsenal” (e.g., greedy heuristics) for computational superiority.

Contributions: Our contributions can be summarized as follows:

- We introduce a new adaptive step size for first-order methods to solve (2), based on the self-concordance property. This technique can be incorporated in many other minimization problems with the same property. Moreover, this tool can be subsumed in many existing schemes [3] with a wide range of diverse regularization terms, decreasing their time-complexity.

This work was supported in part by the European Commission under Grant MIRG-268398, ERC Future Proof, ARO MURI W911NF0910383, and SNF 200021-132548. VC also would like to acknowledge Rice University for his Faculty Fellowship.
• To illustrate the substance of the step size selection, we propose the (F)PS ((F)ast Proximal algorithms for Self-concordant functions) framework and show its computational- and memory-efficiency. The resulting schemes have fast convergence and require the minimum number of input parameters.

2. PRELIMINARIES

Notation: We reserve lower-case and bold lower-case letters for scalar and vector representation, respectively. Upper-case letters denote matrices. The inner product between matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$ is denoted as $\text{tr} (\mathbf{A}^T \mathbf{B})$, where $\text{tr}(\cdot)$ is the trace operator. Given a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, we reserve $\text{diag}(\mathbf{A}) \in \mathbb{R}^{n \times n}$ to denote the diagonal matrix with entries taken from the diagonal of $\mathbf{A}$.

We reserve $\mathbb{R}^+$ to denote the set of positive scalars. Let $\mathbb{S}^n_{++}$ denote the set of positive definite $n \times n$ matrices. For $p (\mathbf{X}) : \mathbb{S}^n_{++} \to \mathbb{R}$, the gradient is denoted as $\nabla p(\mathbf{X})$; for $h(x) : \mathbb{R} \to \mathbb{R}$, we use $h'(x), h''(x)$ to denote the first, second, and third derivative.

Definition 1 (Bregman divergence). Let $p : \mathbb{S}^n_{++} \to \mathbb{R} \cup \{+\infty\}$ be a continuously differentiable and strictly convex function. Given $\Theta_1, \Theta_2 \in \mathbb{S}^n_{++}$, the Bregman divergence $D_p(\cdot \mid \cdot)$ is given by:

$$D_p(\Theta_1 \mid \Theta_2) = p(\Theta_1) - p(\Theta_2) - \text{tr}(\nabla p(\Theta_2)(\Theta_1 - \Theta_2)).$$

Definition 2 (Convexity bounds in gradient methods). Let $p : \mathbb{S}^n_{++} \to \mathbb{R}$ be a strongly convex function with continuous Lipschitz gradient $\nabla p(X)$ for $X \in \mathbb{S}^n_{++}$. Then, there exist $\mu, L > 0$ such that, for any $\Theta_1, \Theta_2 \in \mathbb{S}^n_{++}$.

$$\frac{\mu}{2} \leq D_x(\Theta_1, \Theta_2) \leq \frac{L}{2}.$$  

Proposition 1 (Step size selection for strongly convex gradient descent schemes). For strongly convex (unconstrained) minimization problems using $g(\mathbf{X})$ where $g : \mathbb{R}^{n \times n} \to \mathbb{R}$, $\tau^* := 2/(\mu + L)$ is the optimal step size in the gradient descent scheme $\mathbf{X}_i+1 = \mathbf{X}_i - \tau^* \nabla g(\mathbf{X}_i)$ [15].

Definition 3 (Second order expansion of a function). [16] Let $h : \mathbb{R} \to \mathbb{R}$ be a twice differentiable over an open set $S$. Then, for $x,y \in S$, there exists a constant $\alpha \in [0,1]$ such that:

$$h(x+y) = h(x) + h'(x \cdot y) + \frac{1}{2} y^T h''(x \cdot y) + o(y).$$

(3)

Definition 4 (Self-concordant functions). [17] A convex function $h : \mathbb{R} \to \mathbb{R}$ is self-concordant if $|h''(x)| \leq 2h''(x)^{1/2}, \forall x \in \mathbb{R}$. Given two self-concordant functions $h_1, h_2, h_1 + h_2$ is self-concordant.

Lemma 1 (Upper and lower bounds on second derivatives for self-concordant functions). [17] Let $h : \mathbb{R} \to \mathbb{R}$ be a strictly convex, self-concordant function. Then, $h''(t)$ satisfies:

$$\frac{h''(0)}{1 + t \sqrt{h''(0)}} \leq h''(t) \leq \frac{h''(0)}{1 - t \sqrt{h''(0)}},$$

where both bounds are valid for $0 \leq t < 1/\sqrt{h''(0)}$.

3. GRAPH SELECTION VIA PROXIMAL METHODS

Given that $F(\Theta) := f(\Theta) + g(\Theta)$ is strictly convex and provided a putative solution $\Theta_i \in \mathbb{S}^n_{++}$, an iterative descent scheme follows:

$$\Theta_{i+1} = \Theta_i + \tau^* \Delta,$$

where $\Delta \in \mathbb{R}^{n \times n}$ is a descent direction such that $F(\Theta_{i+1}) < F(\Theta_i)$ for $\tau^* > 0$. To compute $\{\Delta, \tau^*\}$, we can form the following optimization problem:

$$\{\Delta, \tau^*\} = \arg \min_{\Delta \in \mathbb{R}^{n \times n}, \tau \geq 0} \{F(\Theta_i + \tau \Delta) : \Theta_i + \tau \Delta > 0\}. \quad (4)$$

While (4) is the proper way to compute a direction $\Delta$ and a corresponding step size $\tau^*$, in this paper we present an approximation scheme to (4) that introduces the notion of self-concordance in step size selection and performs extremely well in practice; we reserve the detailed convergence analysis for an extended version.

To this end, the proposed algorithm iteratively computes a putative solution by forming a quadratic surrogate only for $f(\Theta)$ at $\Theta_i \in \mathbb{S}^n_{++}$, i.e., $f(\Theta) \leq U(\Theta, \Theta_i) := f(\Theta_i) + \text{tr}(\Delta \cdot (\Theta - \Theta_i)) + \frac{1}{2\tau^*}(\Theta - \Theta_i)^2$, for a carefully selected $\tau^* > 0$ and a direction satisfying $\Delta := -\nabla f(\Theta_i)$, depending only on $f(\cdot)$, i.e., we ignore the presence of $g(\cdot)$ in $F(\cdot)$. Then, instead of minimizing (2), we iteratively solve the following problem:

$$\Theta_{i+1} = \arg \min_{\Theta_i} \left\{U(\Theta_i, \Theta_i) + g(\Theta)\right\}, \quad (5)$$

which can be equivalently stated in proximity operator form [18] as:

$$\Theta_{i+1} = \arg \min_{\Theta_i \geq 0} \left\{ \frac{1}{2\tau^*} \|\Theta - (\Theta_i + \tau^* \Delta)\|_F^2 + g(\Theta) \right\}. \quad (6)$$

The recursive relation in (6) proposes an optimization recipe : given a step size $\tau^*$, we perform a gradient descent step $\Theta_i + \tau^* \Delta$ where $\Delta := -\nabla f(\Theta_i)$ followed by a soft-thresholding operation $\Theta_{i+1} = \text{Soft}(\Theta_i, \tau^* \rho)$ with threshold $\tau^* \rho$ as the closed-form solution the the proximity operator in (6). Finally, we perform a projection onto the positive definite cone using eigenvalue decomposition.

4. $\tau^*$ SELECTION FOR SELF-CONCORDANT FUNCTIONS

Given $\Delta := -\nabla f(\Theta_i)$, we perform a gradient descent step $X_i = \Theta_i - \tau^* \nabla f(\Theta_i)$ where $\tau^* > 0$ and $\nabla f(\Theta_i) := -\Theta_i + \Sigma$. Since $\tau^*$ is unknown, for clarity let $X_i = \Theta_i - \tau \nabla f(\Theta_i)$ where $\tau$ is the unknown variable step size. Then, for $\Theta_1 := X_0$ and $\Theta_2 := \Theta_1$ in Bregman divergence, we define function $\phi(\tau)$ as:

$$\phi(\tau) := D_f(X_i \mid \Theta_i) = -\log \det(X_i) + \log \det(\Theta_i)$$

$$+ \text{tr}(\Theta_i^{-1}(X_i - \Theta_i))$$

$$- \log \det(\Theta_i - \tau \nabla f(\Theta_i)) + \log \det(\Theta_i)$$

$$- \tau \cdot \text{tr}(\Theta_i^{-1} \nabla f(\Theta_i)). \quad (7)$$

In (7), we can rewrite the first $\log \det(\cdot)$ term as [17]:

$$- \log \det(\Theta_i - \tau \nabla f(\Theta_i)) = - \log \det(\Theta_i) - \sum_{j=1}^{n} \log(1 - \tau \lambda_j),$$

where $\lambda_j$ are the eigenvalues of $\Theta_i^{-1/2} \nabla f(\Theta_i) \Theta_i^{-1/2}$. Then:

$$\phi(\tau) = - \sum_{j=1}^{n} \log(1 - \tau \lambda_j) - \tau \cdot \text{tr}(\Theta_i^{-1} \nabla f(\Theta_i)), \quad (8)$$

which is a self-concordant function as the superposition of a self-concordant and a linear (thus self-concordant) function.

Remark 1. In (8), we assume $1 - \tau \lambda_j \geq 0$, $\forall j$ by the definition of the logarithm function. Subsequently, we show that our step size selection always satisfies these conditions, $\forall j$. 

We observe that (8) is strictly convex as a function of \( \tau \). Applying the second order expansion (Definition 3) on \( \phi(\tau) \), we have:

**Lemma 2.** The function \( \phi(\tau) \) satisfies: \( \phi(\tau) = \frac{1}{2} \cdot \tau^2 \cdot \phi''(\hat{\tau}) \), for \( \hat{\tau} \in [0, \tau] \) and \( \phi''(\hat{\tau}) = \sum_{j=1}^{n} \frac{\lambda_j^2}{(1-\tau \lambda_j^2)^3} \).

**Proof.** For \( y := \tau, x := 0 \) and \( a \cdot y := \hat{\tau} \) in Definition 3, the second order expansion of \( \phi(\tau) \) satisfies according to (3):

\[
\phi(\tau) = \phi(0) + \phi'(0) \cdot \tau + \frac{1}{2} \cdot \tau^2 \cdot \phi''(\hat{\tau}).
\]

It is easy to verify the following: (i) \( \phi(0) = 0 \), (ii) \( \phi''(\hat{\tau}) = \sum_{j=1}^{n} \frac{\lambda_j^2}{(1-\tau \lambda_j^2)^3} \). Moreover, \( \phi'(0) = \sum_{i=1}^{n} \lambda_j \cdot \text{tr}(\Theta_i^{-1} \nabla f(\Theta_i)) \). But \( \sum_{j=1}^{n} \lambda_j = \text{tr}(\Theta_i^{-1} \nabla f(\Theta_i)) \). Therefore, \( \phi(0) = 0 \).

Let \( \xi(\tau) := \frac{\phi''(0)}{(1+\tau \phi''(0))} \). Since \( \phi(\cdot) \) is self-concordant and strictly convex, the following inequalities hold true for \( \tau \in [0, \tau] \):

\[
\xi(\tau) \leq \xi(\hat{\tau}) \leq \xi(0) \tag{9}
\]

From Lemma 2, \( \phi''(0) = \sum_{j=1}^{n} \lambda_j^2 \). We know that \( \text{tr}(A^4) = \sum_{j=1}^{n} \lambda_j^4 \) for \( A \in \mathbb{R}^{n \times n} \) where \( \xi_j \) are the eigenvalues of \( A \). Thus, \( \phi''(0) = \sum_{j=1}^{n} \lambda_j^2 = \text{tr}(\Theta_i^{-1} \nabla f(\Theta_i))^2 \).

Given (7), Lemma 2 and \( \|X_i - \Theta_i\|_2^2 \leq 2 \|\nabla f(Y_i)\|_2^2 \):

\[
D_f(X_i \parallel \Theta_i) = \frac{1}{2} \cdot \tau^2 \cdot \phi''(\hat{\tau}) \Rightarrow D_f(X_i \parallel \Theta_i) \leq \frac{1}{2} \|\nabla f(Y_i)\|_2^2 \phi''(\hat{\tau}) \]

Combining the above equation with (9), we locally have:

\[
\frac{\hat{\mu}}{2} \leq \frac{D_f(X_i \parallel \Theta_i)}{\|\nabla f(Y_i)\|_2^2} \leq \frac{\hat{L}}{2} \tag{10}
\]

where \( \hat{L} = \frac{\delta}{(1+\tau \delta)^2} \) and \( \hat{\mu} = \frac{\delta}{(1+\tau \delta)^2} \), for \( \delta := \phi''(0) \) and \( \epsilon := \|\nabla f(Y_i)\|_2^2 \).

By Definition 2, a safe step size selection at the i-th iteration satisfies \( \tau_i^* := \tau = 2(\hat{\mu} + \hat{L}) \) which leads to the following lemma:

**Lemma 3.** At the i-th iteration, the step size \( \tau_i^* = 2/(\hat{\mu} + \hat{L}) \) is determined as \( \tau_i^* = \frac{1}{2} - \frac{1}{\sqrt{\frac{1}{2} + \frac{4}{3}}} \). Moreover, \( \tau_i^* \) is guaranteed to satisfy 0 \( \leq \tau_i^* < \sqrt{\phi''(0)} \), \( \forall i \).

**Proof.** For \( \tau_i^* := \tau = 2/(\hat{\mu} + \hat{L}) \) we obtain:

\[
\tau = \frac{2}{1+\tau \phi''(0)} + \frac{2}{1-\tau \phi''(0)} \Rightarrow \tau = \frac{2(\hat{\mu} + \hat{L})}{\hat{\mu}^2 - \hat{L}^2} = 0 \tag{11}
\]

with roots \( \tau_{\text{min, max}} = \frac{1}{2} \left( -\frac{1}{2} \pm \frac{1}{\sqrt{\frac{1}{2} + \frac{4}{3}}} \right) \). To use the upper bound in (9), the solution \( \tau \) must satisfy 0 \( \leq \tau < 1/\sqrt{\bar{L}} \). We easily observe that \( \tau_{\text{min}} \leq 0 \). For \( \tau_{\text{max}} = \frac{1}{2} \left( -\frac{1}{2} \pm \frac{1}{\sqrt{\frac{1}{2} + \frac{4}{3}}} \right) \) we have: \( \tau_{\text{max}} \geq 0 \) and \( \tau_{\text{max}} \leq \frac{1}{2} \left( -\frac{1}{2} \pm \frac{1}{\sqrt{\frac{1}{2} + \frac{4}{3}}} \right) = \frac{1}{\sqrt{\bar{L}}} \). Since \( \frac{1}{2} + \frac{4}{3} > 0 \), thus \( \tau_i^* := \tau_{\text{max}} \) such that \( \tau_i^* = 2/(\hat{\mu} + \hat{L}) \) and 0 \( \leq \tau_i^* < \frac{1}{\sqrt{\phi''(0)}} \).

**Remark 2.** An alternative step size selection is computed as the minimum root of \( \tau_i^* = \frac{1}{\sqrt{2}} \). While this scheme performs well, it does not exploit the strong convexity of the smooth term.

**Algorithm 1** Proximal algorithm for Self-concordant functions

**Input:** \( \Sigma \geq 0, \rho, \text{MaxIter, tol} \)

**Initialize:** \( \Theta_0 := \text{diag}(\Sigma)^{-1} \)

repeat

1. \( \{\tau, \nabla f(\Theta_i)\} := \text{compute_tau(} \Sigma, \Theta_i) \)

2. \( X_i = \Theta_i - \tau \cdot \nabla f(\Theta_i) \)

3. \( \Theta_{i+1} = \text{Soft}(X_i, \tau \cdot \rho) \)

4. If \( \Theta_{i+1} > 0 \) then continue

5. else repeat steps 2-3 with \( \tau_i^* := \tau_i^*/2 \).

until \( \text{MaxIter} \) is reached or \( \|\Theta_{i+1} - \Theta_i\|_F \leq \text{tol} \)

**Proposition 2.** The step size selection proposed in Lemma 3 satisfies \( 1 - \tau_i^* \lambda_j \geq 0 \), \( \forall j \) in (8).

**Proof.** By construction, we observe that \( \tau_i^* < 1/\sqrt{\phi''(0)} = \frac{1}{\sqrt{\bar{L}}} \) where \( \lambda := [\lambda_1, \ldots, \lambda_n] \).

5. BASIC PROXIMAL ALGORITHM

Algorithm 1 shows the Proximal algorithm for Self-concordant functions (PS) in detail. The per iteration complexity is \( O(n^3) \). The step size selection is dominated by the calculation of the gradient \( \nabla f(\Theta_i) = -\Theta_i^{-1} + \Sigma \); an efficient way to compute \( \Theta_i^{-1} \) is through Cholesky factorization with \( O(n^3) \) complexity. Given \( \nabla f(\Theta_i) \) and \( \Theta_i^{-1} \), the time-complexity for \( \delta := \text{tr}(\Theta_i^{-1} \nabla f(\Theta_i))^2 \) and \( \epsilon := \|\nabla f(\Theta_i)\|_2^2 \) is \( O(n^2) \) while for the quadratic form root-finding step we need \( O(1) \) operations. The soft-thresholding operation requires \( O(n^2) \) complexity.

According to (6), we require \( \Theta_i > 0, \forall i \). The best projection of an arbitrary matrix onto the set of positive definite \( n \times n \) matrices requires an eigenvalue decomposition with \( O(n^3) \) complexity; a prohibitive time-complexity that does not scale well for many applications. In practice though, the projection onto \( S_{++} \) can be avoided with a backtrack line search over \( \tau_i^* \). After soft-thresholding, we can check \( \Theta_{i+1} > 0 \) via its Cholesky factorization. In case \( \Theta_{i+1} \leq 0 \), we decrease the step size \( \tau_i^* := \tau_i^*/2 \) and repeat steps 2 and 3 with complexity \( O(n^2) \). Otherwise, we can reuse the Cholesky factorization of \( \Theta_{i+1} \) to compute \( \Theta_{i+1}^{-1} \) and \( \nabla f(\Theta_{i+1}) \) in the next iteration. In practice though, we rarely need this additional operation.

6. FAST PROXIMAL ALGORITHM

To gain momentum in convergence, we can use memory in estimates as proposed by Nesterov for strongly convex functions [15]; the same acceleration technique has been integrated in other convex approaches and problems such as [11, 19]. Moreover, to overcome the oscillatory behaviour in the trace of the objective value due to the momentum update, we can use adaptive “restart” techniques; c.f. [20]. Algorithm 2 summarizes the FPS scheme; the main difference with Algorithm 1 is that, at each iteration, we no longer operate on the previous estimate \( \Theta_{i-1} \) but rather on \( Y_i \) which simulates an
Algorithm 2: Fast Proximal algorithm for Self-concordant functions

Input: $\Sigma \geq 0, \rho, \text{MaxIter}, \text{tol}$
Initialize: $\Theta_0 = \text{diag}(\Sigma)^{-1}, Y_1 = \Theta_0, \alpha_1 = 1.$
repeat
1. $[\tau_i^*, \nabla f(Y_i, \tilde{\mu}, \tilde{\Sigma})] = \text{compute} \_\text{tau}(\Sigma, Y_i)$ \text{ O}(n^2)
2. $X_i = Y_i - \tau_i^* \nabla f(Y_i)$ \text{ O}(n^2)
3. $\Theta_i = \text{Soft}(X_i, \tau_i^*)$ \text{ O}(n^2)
4. $Y_{i+1} = \Theta_i + \gamma_i (\Theta_i - \Theta_{i-1})$ for $\gamma_i > 0$ \text{ O}(n^2)
5. If $Y_{i+1} > 0$ then continue \text{ O}(1)
6. else repeat steps 2-4 with $\tau_i^* := \tau_i^*/2.$ \text{ O}(n^2)
until MaxIter is reached or $\|Y_{i+1}\| \leq \text{tol}.$

additional (rough) gradient descent step using the previous two estimates $\Theta_i$ and $\Theta_{i-1}.$ To compute $\nabla f(Y_i)$ at each iteration, $Y_i$'s shall satisfy the positive definiteness constraint.

We suggest two schemes for $\gamma_i$ [15]: (A): $\gamma_i = \frac{\alpha_i - 1}{\alpha_i + 1}$ where $\alpha_{i+1} = \frac{1 + 4\alpha_i}{2}$ and $\alpha_1 = 1.$ and (B): $\gamma_i = \frac{1 - \sqrt{1 + \rho \tau_i^*}}{1 + \sqrt{1 + \rho \tau_i^*}}.$ We identified that both strategies perform well in practice where scheme (A) is more stable when $\Sigma$ is rank-deficient (non-strictly convex case).

Since we operate on $Y_i,$ we have to guarantee the positive definiteness of both $\Theta_i$ and $Y_i$ per iteration, leading to an additional Cholesky factorization calculation per iteration. A key lemma for an efficient implementation of Algorithm 2 is the following:

Lemma 4. Given $\Theta_0 > 0, Y_{i+1} > 0$ implies $\Theta_i > 0, \forall i.$

Proof. If $Y_{i+1} > 0,$ then: $\Theta_i > 1 + \gamma_i (\Theta_i - \Theta_{i-1}) > 0 \Rightarrow \Theta_i (1 + \gamma_i) > 1 + \gamma_i \Theta_{i-1} = \Theta_i > 1.$ $\Box$

By Lemma 4, we can check the positive definiteness of $\Theta_i$ through the Cholesky factorization of $Y_{i+1}.$

7. EXPERIMENTS

Experimental configuration: we synthetically generate sparse inverse covariance matrices $\Sigma^{-1},$ according to the simple model:

$$\Sigma^{-1} = I + \Omega, \text{ such that } \Sigma^{-1} > 0 \text{ and } \|\Sigma^{-1}\|_0 = \kappa,$$

where $\Omega \in \mathbb{R}^{n \times n}$ contains random iid off-diagonal entries i.i.d. $\sim N(0, 1).$ Given $\Sigma^{-1},$ we draw $\{X_i\}_{i=1}^p \sim N(0, \Sigma)$ and calculate $\hat{\Sigma}.$ Given the above, we consider two test settings:

(i) $n = 1000, p = n/2$ and $\kappa = 2 \cdot 10^{-3} \cdot n^2.$ To observe interpretable results, we set $p = 5 \cdot 10^{-2}.$

(ii) $n = 3000, p = 5n$ and $\kappa = 10^{-1} \cdot n^2.$ To observe interpretable results, we set $p = 4 \cdot 10^{-2}.$

Linear convergence: We empirically illustrate the convergence rate of the proposed schemes towards a high-accuracy solution $\Theta^*$ of (2); we retain a convergence analysis for an extended version. Let $n = 700, p = 5n,$ $\rho = 2 \cdot 10^{-4}, \kappa = 0.01n^2.$ Figure 1 depicts the linear convergence rate of the proposed schemes and their variants; FPSa uses an adaptive restart scheme [20]. In practice, we observe that the choice of $p$ heavily affects the condition number of the problem and thus the convergence rate of first-order schemes.

8. CONCLUSIONS

Many state-of-the-art gradient approaches for sparse inverse covariance estimation in GMRFs use heuristics to compute a step size which introduce additional “computational losses” due to matrix inversion recalculations or slow convergence. In this work, we present a first-order proximal method which, at its core, utilizes a novel adaptive step size selection procedure based on the self-concordance property of the objective value. Numerical results indicate that our methods outperform state-of-the-art first order methods. Moreover, our framework extends straightforwardly to many convex regularizers; following a simplistic avenue to solve the problem is valuable for the universal application of the algorithm to diverse problems.

Table 1: “Correct”, “Missed” and “Extra” stand for the edges correctly identified, missed or added in the true graph, respectively. MaxIter $= 400$ and tol. $= 10^{-8}.$ “-” depicts no results due to time overhead.

<table>
<thead>
<tr>
<th>Setting (i)</th>
<th>ALM</th>
<th>PS</th>
<th>FPS</th>
<th>FPSa</th>
</tr>
</thead>
<tbody>
<tr>
<td>$|\Theta - \Sigma^{-1}|_F$</td>
<td>0.44 \hspace{1cm} 0.414</td>
<td>0.413 \hspace{1cm} 0.412</td>
<td>0.403 \hspace{1cm} 0.402</td>
<td></td>
</tr>
<tr>
<td>Correct</td>
<td>1705</td>
<td>1893 \hspace{1cm} 1893 \hspace{1cm} 1893</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Missed</td>
<td>291</td>
<td>103 \hspace{1cm} 103 \hspace{1cm} 103</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extra</td>
<td>365</td>
<td>232 \hspace{1cm} 228 \hspace{1cm} 228</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iterations</td>
<td>400</td>
<td>379 \hspace{1cm} 129 \hspace{1cm} 114</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#Inversions</td>
<td>400</td>
<td>379 \hspace{1cm} 129 \hspace{1cm} 114</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Setting (ii): ALM PS FPS FPSa

$\|\Theta - \Sigma^{-1}\|_F$ | - | 0.444 \hspace{1cm} 0.43 | 0.43 |
| Correct | - | 8710 \hspace{1cm} 8725 \hspace{1cm} 8724 | | |
| Missed | - | 290 \hspace{1cm} 275 \hspace{1cm} 276 | | |
| Extra | - | 4 \hspace{1cm} 4 \hspace{1cm} 4 | | |
| Iterations | - | 300 \hspace{1cm} 100 \hspace{1cm} 92 | | |
| #Inversions | - | 300 \hspace{1cm} 100 \hspace{1cm} 92 | | |
9. REFERENCES


