EFFICIENT STOCHASTIC SUBGRADIENT DESCENT ALGORITHMS FOR HIGH-DIMENSIONAL SEMI-SPARSE GRAPHICAL MODEL SELECTION

Songwei Wu, Hang Yu*, and Justin Dauwels*

School of Electrical and Electronic Engineering,
Nanyang Technological University, 50 Nanyang Avenue, Singapore, 639798

ABSTRACT

We consider the structure learning problem of Gaussian graphical models when the underlying graph is semi-sparse. More specifically, we assume that the number of edges in the graph grows quadratically with the dimension P. Similar to the case of sparse graphs, the problem is formulated as maximizing the data log-likelihood with an ℓ_1 norm penalty on the precision matrix (the inverse covariance matrix) that promotes sparsity. We notice that the time complexity of all existing methods is at least $\mathcal{O}(P^3)$ under the scenario of semi-sparse graphs, thus severely hindering their applications to high-dimensional data. By contrast, the time complexity of the proposed method is only $\mathcal{O}(P^2)$ with the help of stochastic gradients. We prove the convergence of the proposed algorithm. Numerical results show that the computational time of the proposed method is shorter than that of the state-of-the-art methods when the graph is semi-sparse.

Index Terms— Graphical Model, Structure Learning, Stochastic Gradient Descent, Gibbs Sampling.

1. INTRODUCTION

Graphical models display the most significant interactions between variables, and can assist the interpretation of complicated systems. In particular, we focus on Gaussian graphical models (GGM) here, in which all variables jointly follow a Gaussian distribution $p(\boldsymbol{x}) \propto \exp\left\{-\frac{1}{2}\boldsymbol{x}^T K \boldsymbol{x} + \boldsymbol{h}^T \boldsymbol{x}\right\}$, where K is the precision matrix and $\boldsymbol{h}$ is the potential vector. Interestingly, the structure of the Gaussian graphical model is characterized by the precision matrix: an edge between node i and j is absent if and only if $K_{ij} = 0$ [1]. Thus, in order to learn the structure of the Gaussian graphical model, we intend to estimate the precision matrix. The resulting problem is formulated as maximizing the log-likelihood of K with an ℓ_1-norm penalty on K (cf. Eq (1)), which encourages sparsity in the off-diagonal entries [2]. The regularization or penalty parameter λ in front of the ℓ_1-norm controls the trade-off between data fidelity and the sparsity of K. If λ is small, then the estimated K tends to be dense; otherwise, K tends to be sparse.

In the sequel, we briefly review some existing deterministic methods for solving the problem. We can divide these methods into 6 categories: (1) block (row/column) coordinate descent methods including GLASSO [1], DP-GLASSO [3], and SINCO [4]; (2) Nesterov's smooth methods and their variant [1, 5, 6]; (3) an inexact primal-dual path-following interior-point algorithm proposed by Li and Toh [7]; (4) augmented Lagrangian methods including ADM[8] and ALM [9], and their variants PSM [10]; (5) proximal first-order methods, such as G-ISTA [11]; (6) proximal Newton methods such as QUIC [12], Newton-Lasso [13], and BIG&QUIC [14].

All the methods are deterministic and the exact gradient of the objective is evaluated in each iteration, which involves the computation of the matrix inverse K^{-1}. The computational complexity of inverting a $P \times P$ matrix is typically $\mathcal{O}(P^3)$. As a result, the time complexity of the majority of the aforementioned methods is at least $\mathcal{O}(P^3)$ [1]-[10]. There are several exceptions though, including QUIC [12], G-ISTA [11], and BIG&QUIC [14]. The time complexity of these methods is $\mathcal{O}(PM)$, where M is the number of nonzero elements in the estimated precision matrix K. More concretely, such methods only update a pruned subset of entries K_{ij} in the precision matrix in each iteration. For instance, QUIC and BIG&QUIC select entries that satisfy either of the following two conditions: (1) $K_{ij} \neq 0$ and (2) $|\nabla_{K_{ij}} f(K)| > \lambda$, where $\nabla_{K_{ij}} f(K)$ is the gradient of the objective function $f(K)$ with regard to (w.r.t.) K_{ij}. When λ is large, the resulting subset has a small size in all iterations, and therefore, the precision matrix is sparse as the algorithms proceed. By exploiting the sparsity of the matrix, the computational complexity of matrix inversion is only $\mathcal{O}(PM)$. However, these methods still suffer from the issue of high computational complexity when the underlying true graph is relatively dense and thus a small λ is preferred. In particular, we introduce the notion of semi-sparse graphs in this paper. In semi-sparse graphs, the number of edges $|\mathcal{E}|$ grows quadratically with the number of nodes $|\mathcal{V}|$, i.e.,

Definition 1. We call the graph family $\{\mathcal{G}_i\}$ as semi-sparse if:

$$\exists c_0 \in \mathbb{R}^+, \forall \mathcal{G}_i \in \{\mathcal{G}_i\} \text{ satisfies } \frac{|\mathcal{E}_i|}{|\mathcal{V}_i| \times (|\mathcal{V}_i| - 1)/2} = c_0.$$

* Both authors contributed equally to the work.

This research is supported by MOE (Singapore) project 2017-T2-2-126.

where $\mathbb{R}^+$ denotes the set of positive real number.

Under this scenario, the time complexity of QUIC, G-ISTA, and BIG&QUIC is still $\mathcal{O}(P^3)$. This issue impedes the application of such methods to high-dimensional semi-sparse graphs.

As a remedy, we propose a novel method by leveraging stochastic subgradients [15] when updating the precision matrix K. Instead of calculating the exact gradient as in the literature, we resort to a noisy but unbiased estimate of the gradient that is computationally cheap to evaluate. In other words, we bypass the computation of the matrix inverse by finding a computationally efficient estimate of it. The resulting time complexity of the proposed algorithm is only $\mathcal{O}(P^2)$, whereas the complexity of the existing methods is a least $\mathcal{O}(P^3)$ when the graph is semi-sparse.

The rest of the paper is organized as follows. In Section 2, we propose a GSGD (Gibbs sampling-based Stochastic subGradient Descent) method for learning sparse precision matrices. In Section 3, we theoretically analyze the convergence of the proposed method. In Section 4, we compare the proposed method with the state-of-the-art methods using both synthetic and real data. Finally, we offer concluding remarks in Section 5.

2. STRUCTURE LEARNING BY STOCHASTIC SUBGRADIENT DESCENT

2.1. Structure Learning of Gaussian Graphical Models

We aim to learn the precision matrix given N multivariate Gaussian distributed observations $\boldsymbol{x}^{(1:N)}$. The resulting optimization problem can be formulated as [1]:

$$\hat{K} \triangleq \underset{K\succ 0}{\operatorname{argmin}} \underbrace{\operatorname{tr}(SK) - \log\det K + \lambda\|K\|_1}_{f(K)}, \qquad (1)$$

where $f(K)$ is the objective function to be minimized, $S = 1/N\sum_{i=1}^{N}\boldsymbol{x}^{(i)}\{\boldsymbol{x}^{(i)}\}^T$ is the empirical or sample covariance matrix, $\|\cdot\|_1$ is the ℓ_1 norm (i.e., the sum of the absolute value of all the elements in the matrix), and λ is the regularization or penalty parameter which balances the tradeoff between data fidelity and model sparsity. One intuitive approach to finding the precision matrix K that minimizes $f(K)$ is to update K along the opposite direction of the subgradient of $f(K)$ w.r.t. K. The exact subgradient can be expressed as:

$$\nabla_K f(K) = S - K^{-1} + \Lambda, \qquad (2)$$

where $\Lambda_{ij} = \operatorname{sgn}(K_{ij})\lambda$, where $\operatorname{sgn}(\cdot)$ is the sign operator. Note that the computational bottleneck in (2) lies in the matrix inversion K^{-1}. As mentioned in Section 1, exact gradients are employed in the aforementioned deterministic methods [1]-[14]. Due to the matrix inversion term in the exact gradient, the time complexity of these methods is $\mathcal{O}(P^3)$ for semi-sparse graphs.

2.2. Stochastic Gradient Descent

To settle the problem of the high time complexity, we exploit stochastic gradients when updating K. Stochastic optimization) [15] iteratively updates K along the direction of an unbiased stochastic estimation of the exact gradient. More precisely, we seek an unbiased estimate $g(K)$ of the exact subgradient $\nabla_K f(K)$, that is, $\mathbb{E}[g(K)] = \nabla_K f(K)$, where $\mathbb{E}[\cdot]$ denotes expectation operation, and refer to $g(K)$ as the stochastic subgradient. It is often computationally cheaper to evaluate the stochastic gradient $g(K)$ than $\nabla_K f(K)$. The stochastic subgradient descent algorithm then proceeds by iteratively following realizations of $-g(K)$ with the step size $\rho_{(\kappa)}$ in iteration κ:

$$K_{(\kappa)} = K_{(\kappa-1)} - \rho_{(\kappa)} g(K_{(\kappa-1)}). \qquad (3)$$

When the step size $\rho_{(\kappa)}$ is set properly, this algorithm is guaranteed to converge to the global minimum of $f(K)$, which will be analyzed in the next section.

Recall that the high computational complexity of computing $\nabla_K f(K)$ in Eq. (2) arises from the matrix inverse K^{-1}. As such, we intend to find an unbiased estimate of K^{-1} that can be efficiently calculated. To this end, we consider $K_{(\kappa)}$ as the precision matrix of a zero-mean Gaussian distribution and draw samples from this Gaussian distribution $\mathcal{N}(0, K_{(\kappa)}^{-1})$. It follows that the covariance of the samples is an unbiased estimate of $K_{(\kappa)}^{-1}$ [16]. Gibbs sampling [17] is exploited to sample from $\mathcal{N}(0, K_{(\kappa)}^{-1})$. Specifically, starting from a random vector $\boldsymbol{y}^{(0)}$, we can sample the ith component of t-th Gibbs sample $y_i^{(t)}$ from the following one-dimensional conditional distribution:

$$\begin{aligned} &p(y_i|y_1^{(t)},\cdots,y_{i-1}^{(t)},y_{i+1}^{(t-1)},\cdots,y_P^{(t-1)}) \\ =&\mathcal{N}\left(\frac{\sum_{j=1}^{i-1}K_{ij}y_j^{(t)}+\sum_{j=i+1}^{P}K_{ij}y_j^{(t-1)}}{K_{ii}},\frac{1}{K_{ii}}\right), \end{aligned}$$

where $(\sum_{j=1}^{i-1}K_{ij}y_j^{(t)}+\sum_{j=i+1}^{P}K_{ij}y_j^{(t-1)})/K_{ii}$ is the conditional mean and $1/K_{ii}$ is the conditional variance. We then cycle through $i = 1,\cdots,P$ until generating L Gibbs samples. The estimate of $K_{(\kappa)}^{-1}$ can be expressed as the covariance of the Gibbs samples:

$$\hat{K}_{(\kappa)}^{-1} = \frac{1}{L}\sum_{t=1}^{L}\boldsymbol{y}_{(t)}{\boldsymbol{y}_{(t)}}^T, \qquad (4)$$

and so the stochastic subgradient can be written as:

$$g(K_{(\kappa)}) \triangleq S - \hat{K}_{(\kappa)}^{-1} + \Lambda. \qquad (5)$$

When setting the number of Gibbs samples L to be invariant w.r.t. the dimension P, the computational complexity of evaluating $g(K_{(\kappa)})$ (5) is only $\mathcal{O}(P^2)$. In our later experiments, a constant $L = 100$ proves applicable. Note that the above procedure spares us from computing $K_{(\kappa)}^{-1}$ deterministically, whose time complexity is $\mathcal{O}(P^3)$.

2.3. Positive-Definiteness of K

In order to check the positive-definiteness of $K_{(\kappa)}$ in (3), we evaluate the smallest eigenvalue of $K_{(\kappa)}$ via the Lanczos process [18, 19, 20]. The time complexity of this approach is only $\mathcal{O}(P^2)$. If the smallest eigenvalue is positive, we proceed to the next iteration; otherwise, we halve the step size $\rho_{(\kappa)}$ until the value is positive.

3. THEORETICAL RESULTS

In this section we analyze the convergence of the proposed algorithm. We will show that the algorithm is guaranteed to converge to the global optimum with a convergence rate of $\mathcal{O}(\ln(\kappa)/\kappa)$. We first start with a lemma on the strong convexity of the objective function $f(K)$,

Lemma 1. *The objective function $f(K)$ in (1) is ξ-strongly convex.*

Note that the negative log-determinant term in (1) is strongly convex and it follows that above lemma holds.

Next, we show that the second-order moments of the stochastic gradients $g(K)$ is finite, that is,

Lemma 2. *$\exists\, G$, so that $\forall\ \kappa$, $\mathbb{E}[\|g(K_{(\kappa)})\|^2] < G^2$.*

We defer the proof to the journal version of this work.

Given Lemma 1 and Lemma 2, we can now analyze the convergence rate of stochastic subgradient descent algorithms according to the following theorem:

Theorem 1. *[21, 22] Suppose that the objective function $f(K)$ is ξ-strongly convex, the stochastic gradients $g(K_{(\kappa)})$ are unbiased, and $\mathbb{E}[\|g(K_{(\kappa)})\|^2] < G^2$ for all κ. Consider stochastic subgradient descent with step size $\rho_{(\kappa)} = c/\kappa$, where c is a positive constant. Then the stochastic subgradient descent algorithm converges to the global optimum of $f(K)$ with convergence rate $\mathcal{O}(\ln(\kappa)/\kappa)$.*

We notice that $g(K_{(\kappa)})$ resulting from the Gibbs sampler is only asymptotically unbiased when $L \to \infty$. In practice, however, a finite L still guarantees the convergence.

On the other hand, instead of setting the stepsize to be c/κ as in Theorem 1, we utilize a dynamic step size scheme, since such schemes are shown to speed up the convergence both theoretically and practically [23, 24]. Concretely, we determine the step size as:

$$\begin{aligned} \rho_{(\kappa)} &= \eta^{\kappa} \frac{\left\|\mathbb{E}[g]^T \mathbb{E}[g]\right\|_{\text{Fro}}^2}{\|\mathbb{E}[g^2]\|_{\text{Fro}}^2}, \\ \mathbb{E}[g^2]_{(\kappa)} &= \beta \mathbb{E}[g^2]_{(\kappa-1)} + (1-\beta) g(K_{(\kappa)})^T g(K_{(\kappa)}), \\ \mathbb{E}[g]_{(\kappa)} &= \beta \mathbb{E}[g]_{(\kappa-1)} + (1-\beta) g(K_{(\kappa)}), \end{aligned} \tag{6}$$

where $\mathbb{E}[g]_{(\kappa)}$ and $\mathbb{E}[g^2]_{(\kappa)}$ respectively represent the first and second order moment of the stochastic gradient in iteration κ and they are approximated by the corresponding moving average, η is a shrinking parameter, and $0 < \beta < 1$ is a weight parameter, and $\|\cdot\|_{\text{Fro}}$ denotes Frobenius norm. Note that the second order moment can be decomposed as $\mathbb{E}[g^2] = \mathbb{E}[g]^T \mathbb{E}[g] + \mathbb{V}[g]$, where $\mathbb{V}[g]$ denotes the variance of the stochastic gradient. If the variance $\mathbb{V}[g]$ is large, the above scheme shrinks the step size, thus mitigating the risk of taking a large step in a wrong direction. Otherwise, the step size becomes large, and the convergence is accelerated. We name the resulting algorithm GSGD (Gibbs sampling-based Stochastic subGradient Descent). The algorithm with default settings of parameters is summarized in Algorithm 1.

Algorithm 1: Gibbs sampling-based Stochastic subGradient Descent (GSGD)

Input : Data $\boldsymbol{x}^{(1:N)}$, penalty parameter λ, number of samples $L = 100$, $\beta = 0.9$, $\eta = 0.99$
Output : K, the sparse precision matrix

```
Initialize the first guess as a diagonal matrix K_(0) satisfying [K_(0)]_ii = (S_ii + λ)^{-1};
while (1) do
    Generate {y_(t) | t = 1, 2 ... L} via Gibbs sampling;
    Compute the subgradient (4), and step size (6);
    while (1) do
        Compute the smallest eigenvalue λ_min of the updated K_new;
        Update K as in (3);
        if λ_min > 0 then
            break;
        else
            halve the step size and update K_new;
        end
    end
    if converged; then
        break;
    end
end
```

4. NUMERICAL RESULTS

In this section, we compare the proposed method with the state-of-the-art methods QUIC [12] and G-ISTA [11].[1] We will analyze both synthetic and real data. All the computations are performed on 64-bit OS Windows server 2012 with two Intel Xeon(R) CPU E5-2690 v2 @3.00GHz processors with 32.0 GB RAM. Note that QUIC and G-ISTA are implemented in C++, while the proposed GSGD method is implemented in MATLAB.

[1] We notice that the methods proposed more recently, such as BIG&QUIC [14] and BCDIC [25], are much slower than QUIC and G-ISTA when applied to semi-sparse graphs. Therefore, we only report the results of QUIC and G-ISTA.

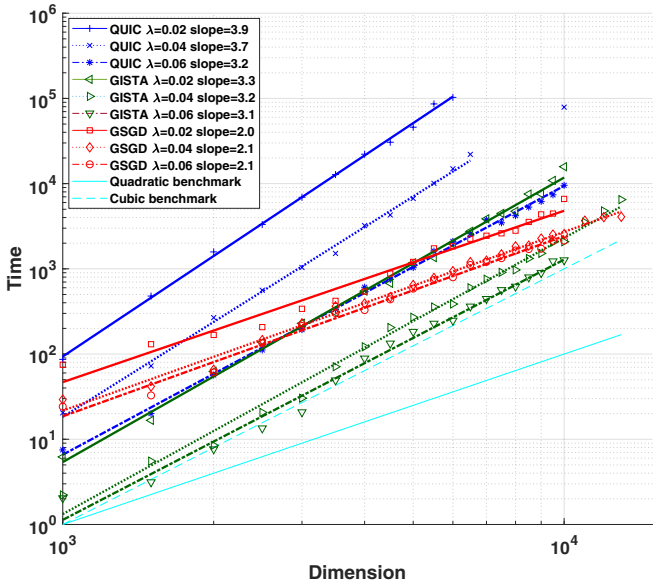


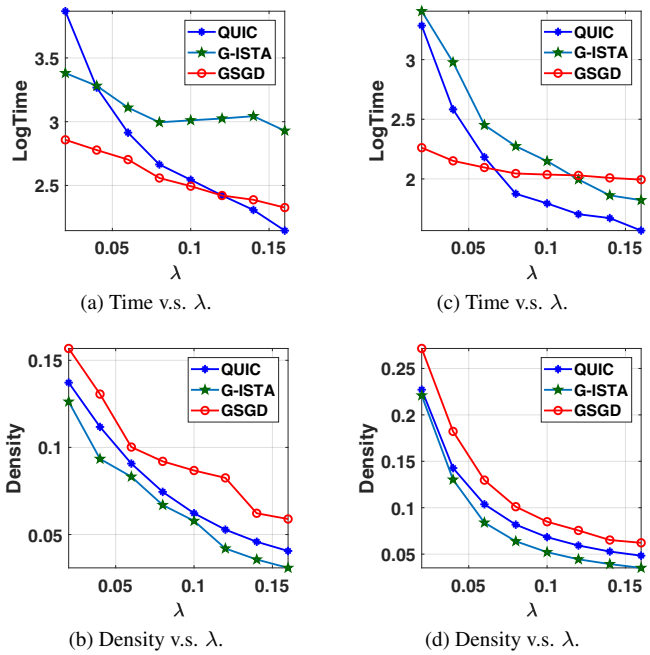
Fig. 1: Computational Time as a function of dimension P for different λ values.

4.1. Synthetic Data

We generate precision matrices with random sparsity pattern and make sure that the number of edges increases quadratically the dimension P. More specifically, We investigate the performance of all methods as the dimension increases from 1,000 to 15,000. The graph density is fixed to be 1% and the sample size is 1,000. The results of computational time as a function of P is shown in Figure 1. Here, we choose $\lambda = \{0.02, 0.04, 0.06\}$ such that the true graph can be well estimated. We fit the computational time by a line to explicitly show the increasing trend. We can find that the computational time of GSGD is approximately a quadratic function of the dimension, regardless of the λ value. However, the computational time of QUIC and G-ISTA is sensitive to the chosen λ, as expected. The slope of the fitted line increases as λ decreases. Moreover, it is approximately a cubic function of P. On the other hand, we also show the mean square error (MSE) between the estimated precision matrix resulting from GSGD and the two benchmark methods in **??**. The MSE is very small, indicating that GSGD achieves the same accuracy with QUIC and G-ISTA but with less amount of computational time, especially for high dimensional cases.

4.2. Real Data

We consider two real gene data sets, including Lukemia data and Estrogen receptor (ER). The first data set was developed for cancer classification originally and it contains 1255 genes monitored by microarrays from 72 samples; the second one, on the other hand, was collected for predicting disease outcome via gene expressions, which consists of 158 samples for 692 genes.



(a) Time v.s. λ. (c) Time v.s. λ.

(b) Density v.s. λ. (d) Density v.s. λ.

Fig. 2: The computational time and density of estimated graphs as a function of λ. The results of Lukemia and ER data are shown respectively on left (a-b) and right (c-d).

In Figure 2, we depict the computational time and the density of the estimated graph as a function of λ. The proposed GSGD becomes faster than QUIC and G-ISTA as λ decreases. More precisely, GSGD outperforms the other two methods in terms of running time when $\lambda = 0.12$ for the data Lukemia and $\lambda = 0.06$ for ER. The corresponding density of the estimated graphs at these λ values are around 10%. In other words, the graphs are still sparse. We emphasize that we prefer a graph that is slightly denser than the ground truth in practice [26], since the false positives can be further removed in future analysis whereas the false negatives are buried by the massive number of true negatives. Hence, the proposed GSGD is favored in practice, since it is able to yield a relatively dense graph with the smallest amount of computational time.

5. CONCLUSION

In this paper, we present a novel method named GSGD to address the structure learning problem of semi-sparse Gaussian graphical models. We prove its convergence and further validate the method on both synthetic and real data.

6. REFERENCES

[1] O. Banerjee, L. El Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data," *The Journal of Machine Learning Research*, vol. 9, pp. 485–516, 2008.

[2] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[3] R. Mazumder and T. Hastie, "The graphical lasso: New insights and alternatives," *Electronic journal of statistics*, vol. 6, p. 2125, 2012.

[4] K. Scheinberg and I. Rish, "Sinco-a greedy coordinate ascent method for sparse inverse covariance selection problem," *preprint*, 2009.

[5] A. d'Aspremont, O. Banerjee, and L. El Ghaoui, "First-order methods for sparse covariance selection," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 1, pp. 56–66, 2008.

[6] Z. Lu, "Smooth optimization approach for sparse covariance selection," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1807–1827, 2009.

[7] L. Li and K.-C. Toh, "An inexact interior point method for l 1-regularized sparse covariance selection," *Mathematical Programming Computation*, vol. 2, no. 3-4, pp. 291–315, 2010.

[8] X. Yuan, "Alternating direction methods for sparse covariance selection," *preprint*, vol. 2, no. 1, 2009.

[9] K. Scheinberg, S. Ma, and D. Goldfarb, "Sparse inverse covariance selection via alternating linearization methods," in *Advances in neural information processing systems*, 2010, pp. 2101–2109.

[10] J. Duchi, S. Gould, and D. Koller, "Projected subgradient methods for learning sparse gaussians," *arXiv preprint arXiv:1206.3249*, 2012.

[11] B. Rolfs, B. Rajaratnam, D. Guillot, I. Wong, and A. Maleki, "Iterative thresholding algorithm for sparse inverse covariance estimation," in *Advances in Neural Information Processing Systems*, 2012, pp. 1574–1582.

[12] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar, "Quic: quadratic approximation for sparse inverse covariance estimation." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2911–2947, 2014.

[13] F. Oztoprak, J. Nocedal, S. Rennie, and P. A. Olsen, "Newton-like methods for sparse inverse covariance estimation," in *Advances in Neural Information Processing Systems*, 2012, pp. 755–763.

[14] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, P. K. Ravikumar, and R. Poldrack, "Big & quic: Sparse inverse covariance estimation for a million variables," in *Advances in neural information processing systems*, 2013, pp. 3165–3173.

[15] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[16] D. A. Harville, "Use of the gibbs sampler to invert large, possibly sparse, positive definite matrices," *Linear algebra and its applications*, vol. 289, no. 1, pp. 203–224, 1999.

[17] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 721–741, 1984.

[18] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. United States Governm. Press Office Los Angeles, CA, 1950.

[19] G. W. Stewart, "A krylov–schur algorithm for large eigenproblems," *SIAM Journal on Matrix Analysis and Applications*, vol. 23, no. 3, pp. 601–614, 2002.

[20] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.

[21] O. Shamir and T. Zhang, "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes." in *ICML (1)*, 2013, pp. 71–79.

[22] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," *arXiv preprint arXiv:1109.5647*, 2011.

[23] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[24] H. Yu and J. Dauwels, "Modeling spatio-temporal extreme events using graphical models," *IEEE Transactions on Signal Processing*, vol. 64, no. 5, pp. 1101–1116, 2016.

[25] E. Treister and J. S. Turek, "A block-coordinate descent approach for large-scale sparse inverse covariance estimation," in *Advances in neural information processing systems*, 2014, pp. 927–935.

[26] H. Liu, K. Roeder, and L. Wasserman, "Stability approach to regularization selection (stars) for high dimensional graphical models," in *Advances in neural information processing systems*, 2010, pp. 1432–1440.