A fast iterative algorithm for high-dimensional differential network

Zhou Tang¹,² · Zhangsheng Yu¹,² · Cheng Wang¹

Received: 19 December 2018 / Accepted: 31 July 2019 / Published online: 17 August 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract
A differential network is an important tool for capturing the changes in conditional correlations under two sample cases. In this paper, we introduce a fast iterative algorithm to recover the differential network for high-dimensional data. The computational complexity of our algorithm is linear in the sample size and the number of parameters, which is optimal in that it is of the same order as computing two sample covariance matrices. The proposed method is appealing for high-dimensional data with a small sample size. The experiments on simulated and real datasets show that the proposed algorithm outperforms other existing methods.

Keywords ADMM · Differential network · Gaussian graphical model · High-dimensional data · Precision matrix

1 Introduction
Covariance matrices that describe the correlations between covariates play an important role in multivariate statistical analysis. For high-dimensional data where the number of covariates is large, it is challenging to estimate the covariance matrix. In the literature, many statistical methods have been proposed to estimate the covari-

Co-first author: Zhangsheng Yu.

chengwang@sjtu.edu.cn
johntan@sjtu.edu.cn
yuzhangsheng@sjtu.edu.cn

¹ School of Mathematical Sciences, MOE-LSC, Shanghai Jiao Tong University, Shanghai 200240, China
² Department of Bioinformatics and Biostatistics, Shanghai Jiao Tong University, Shanghai 20040, China
ance matrix (Bickel and Levina 2008; Rothman et al. 2009; Cai and Liu 2011) or its inverse, which is usually called the precision matrix (Meinshausen and Bühlmann 2006; Jerome et al. 2008; Cai et al. 2011; Zhang and Zou 2014). More details can be found in recent review works by Tong et al. (2014) or Fan et al. (2016).

In this work, we study the covariance structure for the two sample cases. Assume that we have observations from two groups of subjects: $X_1, \ldots, X_{n_1}$ and $Y_1, \ldots, Y_{n_2}$ whose population covariance matrices are $\Sigma_1$ and $\Sigma_2$, respectively. Our interest is to estimate the differential network $\Delta^* = \Sigma_2^{-1} - \Sigma_1^{-1}$, which is the difference between two precision matrices. In biostatistics, the differential network describes the changes in conditional interdependencies between components under different environmental or genetic conditions. See Barabási and Oltvai (2004), Bandyopadhyay et al. (2010), Barabási et al. (2011), Gambardella et al. (2013), Ideker and Krogan (2014), Zhao et al. (2014), Gambardella et al. (2015), Hsiao et al. (2016), Wu et al. (2016), and Kaushik et al. (2017) for examples and the references therein. Another application of the differential network is the quadratic discriminant analysis in multivariate statistical analysis (Anderson 2003). Under the Gaussian distribution assumption, the differential network is exactly the coefficients for the interaction terms between covariates. For quadratic discriminant analysis, it is necessary to recover the differential network (Li and Shao 2015; Jiang et al. 2018).

In recent decades, a large number of statistical methods have been proposed to estimate $\Delta^*$, which can be classified into two categories. The first category separately estimates precision matrices $\Sigma_1^{-1}$ and $\Sigma_2^{-1}$; then taking the difference yields the final estimate for the differential network. The methods for estimating a single precision matrix (Meinshausen and Bühlmann 2006; Jerome et al. 2008; Cai et al. 2011; Zhang and Zou 2014) can be used directly. The second approach jointly estimates precision matrices $\Sigma_1^{-1}$ and $\Sigma_2^{-1}$ (Julien et al. 2011; Jian et al. 2011; Zhu and Li 2018). A joint loss function for the precision matrices is conducted, and we can simultaneously estimate the precision matrices by penalizing the joint loss function. These methods assume that each precision matrix is sparse and can be recovered consistently, which is too strong of an assumption for many applications. Moreover, because our interest is only the differential network, it is not necessary to recover each network for all the subjects.

Recently, Zhao et al. (2014) developed a direct estimator for the differential network $\Delta^*$ under high-dimensional settings. Motivated by Cai et al. (2011), they proposed a Dantzig-type estimator for the high-dimensional differential network. By studying high-dimensional quadratic discriminant analysis, Jiang et al. (2018) proposed a LASSO-type estimator that regularized a convex loss function with an $\ell_1$ penalized term. Usually, the estimators of Zhao et al. (2014) and Jiang et al. (2018) are not symmetric, and a further symmetrical step is needed for the final estimation. Yuan et al. (2017) conducted a one-step symmetric estimator. Under mild conditions, these estimators are all shown to be consistent by assuming that the differential network matrix is sparse. Computationally, they all used the alternating direction method of multipliers (ADMM) (Boyd et al. 2011) to solve the optimization problems. Specifically, Zhao et al. (2014) used a proximal linearization procedure to solve the Dantzig-type optimization problem. The $\ell_1$ penalized problem of Jiang et al. (2018) can be solved by standard ADMM, and Yuan et al. (2017) proposed a two-step ADMM algorithm.
For high-dimensional data where $p \gg n$, the computational complexity of Zhao et al. (2014) is $O(p^4)$, while Jiang et al. (2018) and Yuan et al. (2017) improved the complexity to $O(p^3)$. Recently, Cai and Zhang (2018) established the minimax lower bounds for high-dimensional differential network estimation, and they used a constrained convex optimization algorithm with computational complexity $O(p^3)$. In this paper, we introduce a fast iterative shrinkage-thresholding algorithm (Beck and Teboulle 2009) to minimize the loss functions defined in Yuan et al. (2017) and Jiang et al. (2018). The computational complexity of the new method is improved to approximately $O(np^2)$, which is the same as computing the two sample covariance matrices. Moreover, the proposed iterative shrinkage-thresholding algorithm is a first-order method that is based on gradients and avoids calculating the inverse of matrices. The theoretical convergence rate is also given in this paper. Finally, simulation studies and real data analysis justify the advantages of our algorithm. An R package of our method has been developed and is available at https://github.com/zhoutang776/Diffnet.

The rest of the paper is organized as follows. In Sect. 2, we introduce the loss functions in existing methods and propose a new algorithm. Evaluations in simulated data are presented in Sect. 3, and in Sect. 4, the algorithm is applied to real datasets to demonstrate its performance. The theoretical convergence rate of the algorithm is shown in the “Appendix”.

2 Main results

For any real matrix $A$, we use $\|A\|_2 = \sqrt{\text{tr}(AA^T)}$ to denote its Frobenius norm, $\|A\|$ to denote its spectral norm and $\|A\|_1$ denotes the sum of the absolute values of $A$.

2.1 Existing methods

Our interest is to estimate the differential network $\Delta^* = \Sigma_2^{-1} - \Sigma_1^{-1}$, which is defined as the difference between two precision matrices. Noting

$$\Delta^* = \Sigma_2^{-1} - \Sigma_1^{-1} = \Sigma_1^{-1} (\Sigma_1 - \Sigma_2) \Sigma_2^{-1}, \quad (2.1)$$

we can obtain

$$\text{vec}(\Delta^*) = (\Sigma_2^{-1} \otimes \Sigma_1^{-1}) \text{vec}(\Sigma_1 - \Sigma_2) = (\Sigma_2 \otimes \Sigma_1)^{-1} \text{vec}(\Sigma_1 - \Sigma_2),$$

where $\otimes$ denotes the Kronecker product, and $\text{vec}(\cdot)$ is the vectorization of a matrix. To estimate $\text{vec}(\Delta)$, following LASSO (Tibshirani 1996), we can consider the $\ell_1$ penalized estimation

$$\arg\min \frac{1}{2} \beta^T (S_2 \otimes S_1) \beta - \beta^T \text{vec}(S_1 - S_2) + \lambda \| \beta \|_1,$$
where $S_1$ and $S_2$ are the sample covariance matrices and $\lambda > 0$ is a tuning parameter. Letting $\beta = \text{vec}(\Delta)$, we can obtain the estimate in matrix form

$$
\hat{\Delta}_1 = \arg \min_{\Delta \in \mathbb{R}^{p,p}} \frac{1}{2} \text{tr}\{\Delta^T S_1 \Delta S_2\} - \text{tr}\{\Delta (S_1 - S_2)\} + \lambda \|\Delta\|_1,
$$

which is exactly the estimator proposed by Jiang et al. (2018). Here, the loss function

$$
L_1(\Delta) = \frac{1}{2} \text{tr}\{\Delta^T S_1 \Delta S_2\} - \text{tr}\{\Delta (S_1 - S_2)\},
$$

is convex with respect to $\Delta$, which is appealing for optimization. Generally, the estimate $\hat{\Delta}_1$ is not symmetric, and further symmetrisation is needed to obtain the final estimator. Yuan et al. (2017) considered a symmetric loss function

$$
L_2(\Delta) = \frac{1}{2} \text{tr}\{\Delta^T S_1 \Delta S_2\} + \frac{1}{4} \text{tr}\{\Delta^T S_2 \Delta S_1\} - \text{tr}\{\Delta (S_1 - S_2)\} + \lambda \|\Delta\|_1,
$$

and proposed a symmetric estimation

$$
\hat{\Delta}_2 = \arg \min_{\Delta \in \mathbb{R}^{p,p}} \frac{1}{4} \text{tr}\{\Delta^T S_1 \Delta S_2\} + \frac{1}{4} \text{tr}\{\Delta^T S_2 \Delta S_1\} - \text{tr}\{\Delta (S_1 - S_2)\} + \lambda \|\Delta\|_1.
$$

Theoretically, assuming $\Delta^*$ is sparse, Jiang et al. (2018) and Yuan et al. (2017) showed that $\hat{\Delta}_1$ and $\hat{\Delta}_2$ are consistent estimators for the true differential network $\Delta^*$. Computationally, the loss functions $L_k(\Delta)$ and $k = 1, 2$ are convex functions, and standard ADMM (Boyd et al. 2011) can be used to solve the estimation (2.2) or (2.5). Specifically, for the loss function $L(\Delta) = L_1(\Delta) + L_2(\Delta)$, the augmented Lagrangian function is

$$
L(\Delta, A, B) = L(\Delta) + \rho/2\|\Delta - A + B\|_2^2 + \lambda \|A\|_1,
$$

where $\rho > 0$ is the step size of ADMM. The iterative scheme of ADMM is

$$
\Delta^{k+1} = \arg \min_{\Delta \in \mathbb{R}^{p,p}} L(\Delta, A^k, B^k) = \arg \min_{\Delta \in \mathbb{R}^{p,p}} \rho/2\|\Delta - A^k + B^k\|_2^2,
$$

$$
A^{k+1} = \arg \min_{A \in \mathbb{R}^{p,p}} L(\Delta^{k+1}, A, B^k) = \text{soft}(\Delta^{k+1} + B^k, \lambda/\rho),
$$

$$
B^{k+1} = \Delta^{k+1} - A^{k+1} + B^k,
$$

where soft($A, \lambda$) is an element-wise soft thresholding operator. The $\Delta^{k+1}$ related subproblem dominates the computation of each iteration because the other two subproblems are easy to calculate. Because $L(\Delta)$ is convex, it is equivalent to considering the equation

$$
L'(\Delta) + \rho(\Delta - A^k + B^k) = 0.
$$
For the estimation (2.2), the equation is

\[ S_1 \Delta S_2 - (S_1 - S_2) + \rho(\Delta - A^k + B^k) = 0, \quad (2.6) \]

and solving (2.5) is related to the equation

\[ \frac{1}{2} S_1 \Delta S_2 + \frac{1}{2} S_2 \Delta S_1 - (S_1 - S_2) + \rho(\Delta - A^k + B^k) = 0. \quad (2.7) \]

The Eq. (2.6) can be solved efficiently with the computational complexity \( O(p^3) \), and the explicit solution can be found in Proposition 1 of Jiang et al. (2018) or Lemma 1 of Yuan et al. (2017). For the Eq. (2.7), to derive the explicit solution, it is inevitable to calculate the inverse of a \( p^2 \times p^2 \) matrix with a complexity of \( O(p^4) \). To obtain a computationally efficient algorithm, Yuan et al. (2017) introduced an auxiliary iterative update, which solves the Eq. (2.6) twice and then combines the two solutions. In summary, the computational complexity of Jiang et al. (2018) or Yuan et al. (2017) is \( O(p^3) \), and an eigenvalue decomposition is necessary, which demands high computational memory.

### 2.2 New algorithms

In this paper, we introduce a fast iterative shrinkage-thresholding algorithm (Beck and Teboulle 2009) to solve the penalized estimation (2.2) and (2.5). Compared with ADMM, which needs to solve equations or equivalently calculate the inverse of matrices, the shrinkage-thresholding algorithm is a first-order method that is only based on function values and gradient evaluations. Specifically, for the estimation (2.2) or (2.5), the gradient can be solved efficiently, and then the computational complexity can be improved to \( O(np^2) \) where \( n = n_1 + n_2 \). Under the high dimension small sample size setting where \( p \gg n \), the computational complexity is linear in the sample size and the number of parameters, which is the same as computing the two sample covariance matrices.

For the optimization problem,

\[ \arg \min_{\Delta \in \mathbb{R}^{p \times p}} L(\Delta) + \lambda \| \Delta \|_1, \]

we consider the quadratic approximation at a given point \( \Delta' \in \mathbb{R}^{p \times p} \),

\[ Q(\Delta, \Delta') = L(\Delta') + (\Delta - \Delta')^T \nabla L(\Delta') + \frac{L}{2} \| \Delta - \Delta' \|_2^2 + \lambda \| \Delta \|_1, \quad (2.8) \]

where \( L > 0 \) is the Lipschitz constant for the gradient \( \nabla L(\Delta) \). Because (2.8) is a strongly convex function with respect to \( \Delta \), the unique minimizer of \( Q(\Delta, \Delta') \) for a given \( \Delta' \) is

\[ \arg \min_{\Delta \in \mathbb{R}^{p \times p}} Q(\Delta, \Delta') = \text{soft} \left( \Delta' - \frac{1}{L} \nabla L(\Delta'), \frac{\lambda}{L} \right). \]
Thus, we can solve the optimization problem sequentially

$$\Delta_k = \arg \min_{\Delta} Q(\Delta, \Delta_{k-1}) = \text{soft} \left( \Delta_{k-1} - \frac{1}{L} \nabla L(\Delta_{k-1}), \frac{\lambda}{L} \right).$$

Using the gradient descent algorithm, the sequence \(\{\Delta_k\}\) converges to the solution, and following Beck and Teboulle (2009), we can further use an accelerated scheme to speed up the convergence. Details of the algorithm are summarized in Algorithm 1.

**Algorithm 1** Fast iterative shrinkage-thresholding algorithm for differential network estimation

Input: Lipschitz constant \(L\) of \(\nabla L(\Delta)\) and initial value \(\Delta_0\);

**Step 0.** Start from \(\Delta_{-1} = \Delta_0, t_0 = t_1 = 1\)

**Step 1.** Update

$$\Delta'_k = \Delta_k + \frac{t_{k-1} - 1}{t_k} (\Delta_k - \Delta_{k-1});$$

**Step 2.** Update

$$\Delta_{k+1} = \text{soft}(\Delta'_k - \frac{1}{L} \nabla L(\Delta'_k), \frac{\lambda}{L});$$

**Step 3.** Update

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2};$$

**Step 4.** Repeat Steps 1-3 until convergence.

The main computational burden of this algorithm is Step 2, which involves the multiplication of the matrices. Specifically, we need to calculate the gradients

$$\nabla L_1(\Delta) = S_1 \Delta S_2 - (S_1 - S_2), \quad \nabla L_2(\Delta) = \frac{1}{2} S_1 \Delta S_2 + \frac{1}{2} S_2 \Delta S_1 - (S_1 - S_2),$$

where \(S_1, S_2, \Delta\) are all \(p \times p\) matrices. If we implement the algorithm naively, the computational complexity is \(O(p^3)\), which is the same as that of Jiang et al. (2018) or Yuan et al. (2017). For the high-dimensional data where \(p \gg n\), we have the formulas

$$S_1 = \frac{1}{n_1} X^T X, \quad S_2 = \frac{1}{n_2} Y^T Y,$$

where \(X\) and \(Y\) are the \(p \times n_1\) and \(p \times n_2\) centred data matrices for the two subjects, respectively. Then, the gradient \(\nabla L(\Delta)\) can be calculated efficiently using the facts

$$S_1 \Delta S_2 = \frac{1}{n_1 n_2} X^T (X \Delta Y^T) Y, \quad S_2 \Delta S_1 = \frac{1}{n_1 n_2} Y^T (Y \Delta X^T) X,$$

\(\text{Springer}\)
where the computational complexity can be reduced to $O(np^2)$.

We note that the algorithm is flexible for handling general penalized functions. Specifically, we can consider different tuning parameters for each element, and the new algorithm updates the iterations in Step 2 by using different weights for each soft thresholding operator.

Together with the local linear approximation (Zou and Li 2008), we can extend the algorithms to penalized functions such as MCP or SCAD.

For the fast iterative shrinkage-thresholding algorithm with an accelerated scheme, the sequence of function values $F(\Delta_k) \equiv L(\Delta_k) + \lambda \|\Delta_k\|_1$ can converge to the optimal value $\inf F(\Delta)$ at a linear convergence rate. That is, $F(\Delta_k) - \inf F(\Delta) \leq O(1/k^2)$, which is the best iteration complexity when only first-order information is used (Nesterov 1983). The following theorem gives the $O(\sqrt{L/\epsilon})$ iteration complexity for the Algorithm 1 whose proof is presented in the Appendix for clarity.

**Theorem 2.1** Let $\{\Delta_k\}$ be generated by Algorithm 1 and $\Delta^* = \arg \min F(\Delta)$. Then, for any $k \geq 1$,

$$F(\Delta_k) - F(\Delta^*) \leq \frac{2L\|\Delta_k - \Delta^*\|^2}{(k+1)^2}.$$

### 3 Simulation studies

In this section, we conduct several simulations to demonstrate the performance of the proposed algorithm. In what follows, we refer to the method of Zhao et al. (2014) as “Dantzig”, the ADMM algorithm of Jiang et al. (2018) as “ADMM” and the ADMM algorithm of Yuan et al. (2017) as “sym-ADMM”. The new proposed iterative shrinkage-thresholding algorithm is denoted as “Diffnet” and “sym-Diffnet”. All the algorithms are terminated under the same stop condition $|F(\Delta_k) - F(\Delta_{k+1})| < 10^{-5}(|F(\Delta_k)| + 1)$.

For all of our simulations, we set the sample size $n_1 = n_2 = 200$ and generate the data $X_1, \ldots, X_{n_1}$ and $Y_1, \ldots, Y_{n_2}$ from $N(0, \Sigma_1)$ and $N(0, \Sigma_2)$, respectively. The true differential network is

$$\Delta^* = \Sigma_2^{-1} - \Sigma_1^{-1} = \begin{pmatrix} 0 & -1 & 0 & \cdots & 0 \\ -1 & 2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

and for the precision matrix $\Omega_1 = \Sigma_1^{-1}$, we consider two covariance structures:

- Sparse case: $\Omega_1 = (0.5|i-j|)^{-1}_{p \times p}$. Specifically, $\{\Omega_1\}_{1,1} = \{\Omega_1\}_{p,p} = \frac{4}{3}$ and $\{\Omega_1\}_{i,i} = \frac{5}{3}$ for all other $i$. $\{\Omega_1\}_{i,i+1} = \{\Omega_1\}_{i-1,i} = \frac{2}{3}$ and $\{\Omega_1\}_{i,j} = 0$ for all other $i, j$;
- Asymptotic sparse case: $\Omega_1 = (0.5|i-j|)_{p \times p}$. 

\( \odot \) Springer
Table 1 summarizes the computational time in seconds based on 10 replications where all methods are implemented in R with a PC with a 2.50 GHz Intel Core i5-2450 M CPU and 8 GB of memory. For all the methods, we solve a solution path corresponding to 50 lambda values ranging from $\lambda_{\text{max}}/2$ to $\lambda_{\text{max}}$, where $\lambda_{\text{max}}$ is the maximum absolute element of the differential sample covariance matrices $S_1 - S_2$ corresponding to the estimation $\hat{\Delta} = 0$. From Table 1, we can see that for large $p$, our proposed algorithms are much faster than the original ADMM methods with a complexity of $O(p^3)$ and the “Dantzig” method with a complexity of $O(p^4)$. Specifically, based on ADMM, solving the symmetric estimation (2.5) is slower than calculating the estimation (2.2) because “sym-ADMM” needs to solve the equation (2.6) twice while “ADMM” only needs to calculate (2.6) once. For the proposed shrinkage-thresholding algorithm, we can see that calculating the symmetric estimation uses less time, which means that the symmetry property helps us obtain a faster convergence rate. To further verify this issue, we report the number of iterations in Table 2, from which we can see that “sym-ADMM” does use fewer iterations until convergence.

Figure 1 shows the solution paths of the symmetric estimation for the sparse case and the asymptotic sparse case with different data dimensions $p$. We can see that the $\ell_1$ penalized methods (2.5) can recover the support of the differential network when the tuning parameter is suitably chosen.

4 Real data analysis

In this section, we apply our algorithm to three real datasets.
Table 2  The average number of iterations (standard deviation) for solving a solution path

|       | p = 100  | p = 200  | p = 400  | p = 600  | p = 800  |
|-------|----------|----------|----------|----------|----------|
| Sparse case                |
| Dantzig          | 1104.2 (118.2) | –        | –        | –        | –        |
| ADMM            | 533.8 (50.01)  | 832.4 (129.95) | 1422.4 (193.56) | 1754.4 (231.38) | 3117 (103.74) |
| sym-ADMM        | 1899.6 (211.67) | 2598.6 (386.85) | 4262.6 (556.50) | 5175.5 (903.80) | 6351.6 (431.13) |
| Diffnet         | 726.6 (66.87)  | 757.2 (73.14)  | 1024.8 (68.30)  | 1317.2 (123.99) | 1419.4 (133.34) |
| sym-Diffnet     | 714.2 (78.12)  | 755.4 (74.38)  | 957.8 (74.77)   | 1274.6 (135.413)| 1350 (143.03)   |
| Asymptotic sparse case |
| Dantzig          | 876.5 (67.2)   | –        | –        | –        | –        |
| ADMM            | 818.4 (116.55) | 1413.8 (183.68) | 2560 (201.52)  | 3732.2 (189.24) | 4455.1 (239.72) |
| sym-ADMM        | 2980.2 (569.15) | 4488.6 (520.56) | 6859.2 (1228.12) | 8291.3 (1510.22) | 9458 (1710.22) |
| Diffnet         | 968 (73.78)    | 1113 (90.88)  | 1160.8 (101.89) | 1228.2 (128.59) | 1468.4 (150.33) |
| sym-Diffnet     | 917.6 (88.19)  | 1040.8 (77.89) | 1089.2 (95.94)  | 1158.2 (141.64) | 1388.2 (182.50) |

Fig. 1 The solution paths for different data dimensions $p$ where the top panels are the results for sparse cases and the bottom panels are the results for asymptotic sparse cases.
4.1 Spambase dataset

In this example, we model the differential network of spam and non-spam emails. The data are publicly available at [https://archive.ics.uci.edu/ml/datasets/spambase](https://archive.ics.uci.edu/ml/datasets/spambase), which includes 1813 spam emails and 2788 non-spam emails. The dataset includes 56 attributes, including the frequency of the words and the characters and the length of the uninterrupted sequences of capital letters. More details can be found on the website.

We standardize the data and use a non-paranormal transformation to relax the assumption of Gaussian distribution. Figure 2 illustrates the estimator given by our algorithm where each node represents a specific feature. Our method indicates the existence of several hub features, including “direct”, “telnet”, “technology”, “labs” and “hp”. Therefore, covariance structure changes between spam and non-span emails might exist. For example, because the data were donated by Hewlett-Packard Labs, the words “telnet”, “hp” and “technology” have a higher frequency in non-spam emails, which means these features can help researchers label the emails.

4.2 Hepatocellular carcinoma dataset

As a second example, we apply our algorithm to mRNA expression data of liver cancer patients from the International Cancer Genome Consortium, which is available at [https://icgc.org/icgc/cgp/66/420/824](https://icgc.org/icgc/cgp/66/420/824). Several pathways from the KEGG pathway database were studied. To determine the conditional dependency relationships between liver cancers and normal patterns.

To address the original data, we performed three steps. First, we constrained the mRNAs in the following pathway: pathways in cancer (05200), transcriptional mis-
A fast iterative algorithm for high-dimensional...

Fig. 3 The differential network for the hepatocellular carcinoma dataset

regulation in cancer (05202), viral carcinogenesis (05203), chemical carcinogenesis (05204), proteoglycans in cancer (05205), microRNAs in cancer (05206), central carbon metabolism in cancer (05230), choline metabolism in cancer (05231), and hepatocellular carcinoma (05225). Second, we used the impute function from the R impute package to fill the missing values. Third, we standardize the data and used a non-paranormal transformation, which resulted in 223 liver cancer patients and 222 normal patients with a total of 1209 mRNAs.

Figure 3 summarizes the estimation given by our algorithm, where each node represents a specific mRNA. This figure shows that real transcription networks often contain hub nodes. Our method indicates that SSX1 is an important mRNA. Indeed, SSX1 is a valid treatment option for CTNNB1 mutation-positive HCC patients, while CTNNB1 is a major mutation. Moreover, SSX1 as an oncogene, is functionally validated (Ding et al. 2014).

4.3 Single-cell sequencing dataset

MicroRNA plays an important role in suppressing multiple target genes within a cell population. How it affects transcriptional heterogeneity and gene co-expression remains unknown. As a third example, we apply our algorithm to single-cell sequencing combined with the introduction of individual microRNAs (Gambardella et al. 2017). There are two microRNAs having opposing effects on transcriptional heterogeneity within the cell population, with let-7 increasing and miR-294 decreasing the heterogeneity between cells. This dataset, which is available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE80168, contains wild-type and Dgcr8 knockout cells that were mock-transfected or transfected with miRNA mimics for let-7 or miR-294.
To address the raw data, we performed three procedures. First, we established sample and gene filters based on the number and diversity of reads and evidence of introduced miRNA functions. For sample filters, we used the samples with more than 500 K read counts. For gene filters, we retained the genes where the average read counts were greater than 5. Second, we conducted a Wilcoxon signed-rank test with a significance level $\alpha = 0.01$ to identify the important genes. Third, we standardized the data and used a non-paranormal transformation, which resulted in 55 let-7 cell samples and 76 miRNA cell samples with a total of 3055 genes.

Figure 4 summarizes the estimators with different tuning parameters $\lambda$, where each node is a gene. We can see that more nodes are connected when the regularization parameter $\lambda$ decreases. These connected genes can provide us with more information about how let-7 and miR-294 affect transcriptional heterogeneity and gene co-expression.

5 Conclusions

We propose a fast iterative algorithm to solve differential network estimation, which has $O(np^2)$ time complexity and $O(p^2)$ memory complexity. This time complexity of each iteration is linear in the sample size and the number of parameters, which is the same as computing two sample covariance matrices. One extension of our proposal considers the differential network based on a more general graph network (Liu et al. 2009; Xue and Zou 2012). We also note here that in the present context, we only focus on calculating the solution path. For applications, the selection of tuning parameters is also crucial. The common cross-validation procedure or model selection rules, such as AIC or BIC, are possible for efficiently selecting a good tuning parameter. Further research on these topics is warranted.

Acknowledgements We thank two reviewers, an associate editor, and the editor for their most helpful comments. Yu was supported in part by National Natural Science Foundation of China 11671256, 2016YFC0902403 of the Chinese Ministry of Science and Technology and Neil Shen’s SJTU Medical Research Fund. Wang was partially supported by the Shanghai Sailing Program 16YF1405700 and the National Natural Science Foundation of China 11701367 and 11825104.
Appendix

According to the main results of Beck and Teboulle (2009), to complete the proof of Theorem 1, we need to show only that the loss function $L_1(\Delta)$ is convex, which is the result of the following lemma.

**Lemma 5.1** The loss function (2.3) is a smooth convex function, and its gradient is Lipschitz continuous with Lipschitz constant $L = \lambda_{\text{max}}(S_1)\lambda_{\text{max}}(S_2)$, that is

$$
\|\nabla L_1(\Delta_1) - \nabla L_1(\Delta_2)\|_2 \leq L\|\Delta_1 - \Delta_2\|_2,
$$

where $\lambda_{\text{max}}(S_i)$ is the largest eigenvalue of the sample covariance matrix $S_i$ for $i = 1, 2$.

**Proof** Because the loss function (2.3) is defined by

$$
L_1(\Delta) = \frac{1}{2}\text{tr}\{\Delta^T S_1 \Delta S_2\} - \text{tr}\{\Delta (S_1 - S_2)\},
$$

We can calculate the gradient of $L_1(\Delta)$

$$
\nabla L_1(\Delta) = S_1 \Delta S_2 - (S_1 - S_2),
$$

and the Hessian matrix is $S_2 \otimes S_1$. Because both covariance matrices $S_1$ and $S_2$ are definite positive matrices, the Hessian matrix is a definite positive matrix. Hence, the loss function $L_1(\Delta)$ is a smooth convex function.

Moreover, for any $\Delta_1, \Delta_2 \in \text{dom}(\nabla L_1)$, we have

$$
\|\nabla L_1(\Delta_1) - \nabla L_1(\Delta_2)\|_2 = \|S_1(\Delta_1 - \Delta_2)S_2\|_2
\leq \lambda_{\text{max}}(S_2 \otimes S_1)\|\text{vec}(\Delta_1 - \Delta_2)\|_2
\leq \lambda_{\text{max}}(S_1)\lambda_{\text{max}}(S_2)\|\Delta_1 - \Delta_2\|_2.
$$

The proof is now complete. □

References

Anderson T (2003) An introduction to multivariate statistical analysis. Wiley series in probability and statistics. Wiley, New York

Bandyopadhyay S, Mehta M, Kuo D, Sung MK, Chuang R, Jaehnig EJ, Bodenmiller B, Licon K, Copeland W, Shales M et al (2010) Rewiring of genetic networks in response to DNA damage. Science 330(6009):1385–1389

Barabási AL, Oltvai ZN (2004) Network biology: understanding the cell’s functional organization. Nat Rev Genet 5(2):101

Barabási AL, Gulbahce N, Loscalzo J (2011) Network medicine: a network-based approach to human disease. Nat Rev Genet 12(1):56

Beck A, Teboulle M (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J Imaging Sci 2(1):183–202
Bickel PJ, Levina E (2008) Regularized estimation of large covariance matrices. Ann Stat 36(1):199–227
Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via
the alternating direction method of multipliers. Found Trends Mach Learn 3(1):1–122
Cai T, Liu W (2011) Adaptive thresholding for sparse covariance matrix estimation. J Am Stat Assoc
106(494):672–684
Cai T, Zhang L (2018) A convex optimization approach to high-dimensional sparse quadratic discriminant
analysis. Ann Stat (submitted)
Cai T, Liu W, Luo X (2011) A constrained \( \ell_1 \) minimization approach to sparse precision matrix estimation.
J Am Stat Assoc 106(494):594–607
Ding X, Yang Y, Han B, Du C, Xu N, Huang H, Cai T, Zhang A, Han ZG, Zhou W, Chen L (2014) Transcrip-
tomic characterization of hepatocellular carcinoma with cttnb1 mutation. PLoS ONE 9(5):e95307
Fan J, Liao Y, Liu H (2016) An overview of the estimation of large covariance and precision matrices.
Econom J 19(1):C1–C32
Gambardella G, Moretti MN, De Cegli R, Cardone L, Peron A, Di Bernardo D (2013) Differential network
analysis for the identification of condition-specific pathway activity and regulation. Bioinformatics
29(14):1776–1785
Gambardella G, Peluso I, Montefusco S, Bansal M, Medina DL, Lawrence N, Bernardo DD (2015) A
reverse-engineering approach to dissect post-translational modulators of transcription factors activity
from transcriptional data. BMC Bioinform 16(1):279
Gambardella G, Carissimo A, Chen A et al (2017) The impact of micrornas on transcriptional heterogeneity
and gene co-expression across single embryonic stem cells. Nat Commun 8:14126
Hsiao TH, Chiu YC, Hsu PY, Lu TP, Lai LC, Tsai MH, Huang THM, Chuang EY, Chen Y (2016) Differential
network analysis reveals the genome-wide landscape of estrogen receptor modulation in hormonal
cancers. Sci Rep 6:23035
Ideker T, Krogan NJ (2014) Differential network biology. Mol Syst Biol 8(1):565
Jian G, Elizaveta L, George M, Ji Z (2011) Joint estimation of multiple graphical models. Biometrika
98(1):1–15
Jiang B, Wang X, Leng C (2018) A direct approach for sparse quadratic discriminant analysis. J Mach
Learn Res 19(31):1–37
Julien C, Yves G, Christophe A (2011) Inferring multiple graphical structure. Stat Comput 21(4):537–553
Kaushik A, Ali S, Gupta D (2017) Altered pathway analyzer: a gene expression dataset analysis tool for
identification and prioritization of differentially regulated and network rewired pathways. Sci Rep
7:40450
Li Q, Shao J (2015) Sparse quadratic discriminant analysis for high dimensional data. Stat Sin 25:457–473
Liu H, Lafferty J, Wasserman L (2009) The nonparanormal: semiparametric estimation of high dimensional
undirected graphs. J Mach Learn Res 10:2295–2328
Meinshausen N, Bühlmann P (2006) High-dimensional graphs and variable selection with the lasso. Ann
Stat 34(3):1436–1462
Nesterov Y (1983) A method for solving the convex programming problem with convergence rate \( O(k^2) \).
Soviet Math Dokl 27:372–376
Rothman AJ, Levina E, Zhu J (2009) Generalized thresholding of large covariance matrices. J Am Stat
Assoc 104(485):177–186
Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc Ser B 58(1):267–288
Tong T, Wang C, Wang Y (2014) Estimation of variances and covariances for high-dimensional data: a
selective review. Comput Stat 6(4):255–264
Wu MY, Zhang XF, Dai DQ, Le OY, Zhu Y, Yan H (2016) Regularized logistic regression with network-based
pairwise interaction for biomarker identification in breast cancer. BMC Bioinform 17(1):108
Xue L, Zou H (2012) Regularized rank-based estimation of high-dimensional nonparanormal graphical
models. Ann Stat 40(5):2541–2571
Yuan H, Xi R, Chen C, Deng M (2017) Differential network analysis via the lasso penalized D-trace loss.
Biometrika 104(4):755–770
Zhang T, Zou H (2014) Sparse precision matrix estimation via lasso penalized D-trace loss. Biometrika
101(1):103–120
Zhao SD, Cai TT, Li H (2014) Direct estimation of differential networks. Biometrika 101(2):253–268
Zhu Y, Li L (2018) Multiple matrix gaussian graphs estimation. J R Stat Soc Ser B 80:927–950
Zou H, Li R (2008) One-step sparse estimates in nonconcave penalized likelihood models. Ann Stat 36(4):1509–1533

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.