DIFFERENTIAL ANALYSIS FOR NETWORKS OBEYING CONSERVATION LAWS

Anirudh Rayas[*], *Rajasekhar Anguluri*[*], *Jiajun Cheng*[†], *and Gautam Dasarathy*[*]

[*]School of Electrical, Computer, and Energy Engineering
[†]School of Computing and Augmented Intelligence
Arizona State University, Tempe, AZ, 85281 USA
{ahrayas, rangulur, ccheng58, gautamd}@asu.edu

ABSTRACT

Networked systems that occur in various domains, such as electric networks, the brain, and opinion networks, are known to obey conservation laws. For instance, electric networks obey Kirchoff's laws, and social networks obey opinion consensus. Conservation laws are often modeled as balance equations that relate appropriate injected flows and potentials at the nodes of the networks. A recent line of work considers the problem of estimating the unknown structure of such networked systems from observations of node potentials (and only the knowledge of the statistics of injected flows). Given the dynamic nature of the systems under consideration, an equally important task is estimating the change in the structure of the network from data – the so called *differential network analysis* problem. That is, given two sets of node potential observations, the goal is to estimate the structural differences between the underlying networks. We formulate this novel differential network analysis problem for systems obeying conservation laws and devise a convex estimator to learn the edge changes directly from node potentials. We derive conditions under which the estimate is unique in the high-dimensional regime and devise an efficient ADMM-based approach to perform the estimation. Finally, we demonstrate the performance of our approach on synthetic and benchmark power network data.

Index Terms**—** differential network analysis, structure learning, sparsity, convex optimization, ADMM.

1. INTRODUCTION

A networked system is said to obey a conservation law if flows are neither created nor destroyed. Depending on the context, flows could represent current in electric circuits, water in hydraulic networks, or opinion dynamics in social networks [1]. Such systems are at the heart of many natural, engineering, and societal networks [2]. These laws can be conveniently modeled as balance equations that posit a linear map between injected flows and potentials at the network nodes. For finite-dimensional networks, this linear map is the Laplacian matrix whose sparsity pattern encodes the network structure—the edge connectivity of the network.

In many practical problems of interest, one often does not know the structure of the network, a key information for learning, leveraging, and operating complex systems. Consequently, a recent line of work (see, for example, [3, 4, 5, 6, 7]) considers estimating the network structure from observations of node potentials (and only the knowledge of the statistics of injected flows). Given the dynamic nature of the systems under consideration, an equally important task is estimating the change in the structure of network from data. This problem, dubbed *differential network analysis*, appears in many biological and genomics networks [8, 9, 10, 11] and is the focus of the paper.

The differential network analysis problem we consider is for systems obeying conservation laws and is stated as follows: given node potential observations from a system at two different time instants, estimate the sparse changes in the network at these time instants. A generalized version of this problem is to estimate the sparse changes in two systems using two sets of node potential observations, one from each system. We distinguish it from the existing differential network analysis in that we exploit the relationship between the injected flows and the potentials.

A naïve approach to learning sparse changes is first estimating the individual network structures and then looking for differences in the estimates. Unsurprisingly, such an indirect approach would be statistically inefficient since it expends effort on estimating parameters that are irrelevant to the task at hand (e.g., edges that do remain

This work was supported in part by the National Science Foundation (NSF) under the grants CCF-2048223, CCF-2007688, and CBET-2200161, and by the National Institutes of Health (NIH) under the grant 1R01GM140468-01. Jiajun Cheng was supported by the Arizona State University Summer Research Initiative (SURI) program.

unchanged). To overcome these issues, we propose an ℓ_1-norm regularized convex estimator to learn the sparse edge changes directly using samples from the node potentials. Our estimator exploits the fact that the sparsity pattern of the network is encoded in the *square root* of the inverse covariance matrix of the node potential vector. We derive conditions under which the estimate is unique in the high-dimensional regime. Finally, we present an ADMM approach to numerically solve the estimator and evaluate the performance of our approach on synthetic and benchmark power network data.

2. PRELIMINARIES AND BACKGROUND

Let $\mathcal{G} = (V, E)$ be an undirected connected graph on the node set $V \triangleq \{0, 1, 2, \ldots, p\}$ and edge set $E \subset V \times V$. To each edge (i, j) we associate a non-negative weight $a_{i,j}$. Let X and Y be $p + 1$-dimensional real-valued vectors of injections (in-flows) and potentials (out-flows) at the nodes. Then the basic conservation law between these vectors is $X - B^*Y = 0$, where B^* is a Laplacian matrix such that $B_{i,j} = -a_{i,j}$ for $i \neq j$ and $B_{i,i} = -\sum_{j,i\neq j} a_{i,j}$ for $i = j$. The key property of B^* is that edge $(i, j) \in E$ if and only if $B_{i,j} \neq 0$. The model above is sometimes referred to as a generalized Kirchoff's law. It is flexible enough to describe the relationship between flows and potentials in a variety of systems, including electrical circuits, hydraulic networks, opinion consensus in social networks, etc (see e.g., [4, 1] and references therein). We work with the reduced graph obtained by deleting the node 0 and its edges in $\mathcal{G}$. This reduction is standard in many problems (see e.g., [5, 12]). With an abuse of notation, we denote the Laplacian of the reduced graph as B^*. Importantly, B^* is a $p \times p$ positive definite matrix; and hence, invertible [12]. The invertiblity assumption ensures that B^* is identifiable from Y.

2.1. Differential network analysis

Consider two networked systems $\mathcal{G}_1$ and $\mathcal{G}_2$ with same node sets but different edge sets. Let $X_1 \sim \mathcal{N}(0, \Sigma_{X_1})$ and $X_2 \sim \mathcal{N}(0, \Sigma_{X_2})$ be the injection vectors at the nodes of $\mathcal{G}_1$ and $\mathcal{G}_2$. Then, the corresponding node potentials $Y_i = (B_i^*)^{-1} X_i$ satisfy $Y_i \sim \mathcal{N}(0, {\Theta^*}_i^{-1})$ where $\Theta_i^* = B_i^* \Sigma_{X_i}^{-1} B_i^*$ and $i \in \{1, 2\}$. We model injections as random vectors to account for unmodelled injections in the system. For example, these injections could be instantaneous consumer demands in power networks.

Our objective is to infer the changes in the network structure. Formally, suppose that we have access to n_i i.i.d samples from Y_i. Then, our main goal is to estimate $\Delta_B^* = B_2^* - B_1^*$. This difference matrix captures changes in the edge weights of $\mathcal{G}_1$ and $\mathcal{G}_2$. Of particular interest is the sparsity pattern of Δ_B^* as it indicates how similar (or dissimilar) the network systems are.

We next develop an expression for Δ_B^* as a function of Θ_1^* and Θ_2^* which is a starting point for our algorithm design and analysis. We assume that the injection covariances $\Sigma_{X_i} \succ 0$ are known (see Remark 1 for relaxing this assumption). We recall an important fact that any positive (semi) definite matrix has a unique square root that is also a positive (semi) definite matrix. That is, for any $C \succeq 0$, there exists a unique $M \succeq 0$ such that $C = M^2$ [13].

Consider $\Sigma_{X_i} = M_{X_i}^2$ such that $M_{X_i} \succ 0$ is unique. Define $\widetilde{Y}_i = M_{X_i} Y_i$ and set $\widetilde{\Theta}_i^* = (\mathrm{Cov}[\widetilde{Y}_i])^{-1}$. Then

$$\Delta_B^* = M_{X_2} (\widetilde{\Theta}_2^*)^{\frac{1}{2}} M_{X_2} - M_{X_1} (\widetilde{\Theta}_1^*)^{\frac{1}{2}} M_{X_1}, \qquad (1)$$

where $(\widetilde{\Theta}_i^*)^{\frac{1}{2}} = M_{X_i}^{-1} B_i^* M_{X_i}^{-1} \succ 0$ is the unique square root matrix of $\widetilde{\Theta}_i^*$. The expression in (1) follows by direct substitution. The uniqueness is because $(\widetilde{\Theta}_i^*)^{-1} = M_{X_i} {\Theta^*}_i^{-1} M_{X_i} = (M_{X_i} {B_i^*}^{-1} M_{X_i})^2 \succ 0$.

Because Σ_{X_i} are known, their square roots M_{X_i} are known. Therefore, a natural estimator for Δ_B^* is to replace $(\widetilde{\Theta}_i^*)^{\frac{1}{2}}$ in (1) with its sample estimate—the square root of the inverse of the sample covariance matrix of $\widetilde{Y}_i$. This estimate is unfortunately highly sample inefficient. In fact, the sample covariance matrix is non-invertible when $p > n_i$ (the so-called high-dimensional regime). Alternatively, we can estimate $(\widetilde{\Theta}_i^*)^{\frac{1}{2}}$ using well-known estimators such as GLASSO or CLIME [14, 15, 16]. But these estimators work well only when Θ_i^* is sparse. As shown in our prior work [4], Θ_i^* need not be sparse even when B_i^* is sparse.

We overcome the challenges above by directly estimating Δ_B^* assuming it is sparse. This assumption is mild compared to the stringent assumption that B_i^* is sparse and it is satisfied in several applications (see Introduction).

Remark 1. *(Unknown covariance matrix Σ_{X_i}.)* If Σ_{X_i} is unknown, we can slightly modify Δ_B^* to estimate differences between the square roots $(\Theta_1^*)^{\frac{1}{2}}$ and $(\Theta_2^*)^{\frac{1}{2}}$ (given in (1)). This approach works best if the sparsity of B_i^* (approximately) equals the sparsity of $(\Theta_1^*)^{\frac{1}{2}}$, which for instance happens when Σ_{X_i} is (approximately) diagonal.

3. A CONVEX SQUARE ROOT ESTIMATOR

We introduce our square root difference estimator to estimate a sparse Δ_B^*. Let $\widetilde{\Psi}_i = M_{X_i}^{-1} (\widetilde{\Theta}_i^*)^{-\frac{1}{2}} M_{X_i}^{-1}$, where $(\widetilde{\Theta}_i^*)^{-\frac{1}{2}} = (\mathrm{Cov}[\widetilde{Y}_i])^{\frac{1}{2}}$ as defined in (1). Let $\Delta \in \mathbb{R}^{p \times p}$ and consider the following loss function:

$$\mathcal{L}(\Delta) = \frac{1}{4}(\langle \widetilde{\Psi}_1 \Delta, \Delta \widetilde{\Psi}_2 \rangle + \langle \widetilde{\Psi}_2 \Delta, \Delta \widetilde{\Psi}_1 \rangle) - \langle \Delta, \widetilde{\Psi}_1 - \widetilde{\Psi}_2 \rangle, \qquad (2)$$

where $\langle A, B\rangle \triangleq \mathrm{tr}(AB^{\mathsf{T}})$. Such loss functions, dubbed D-trace losses, have emerged as a computationally efficient alternative to the log-det loss function and are related to score-matching losses [17]. Understanding statistical properties of estimators based on D-trace loss functions is an active study of research [10, 18, 19].

A loss-function similar to (2) has been used in [10] to learn the difference between two graphical models using the covariance matrices. Instead, we learn the difference between two networks using the square roots of the covariance matrices. Using the arguments in [10], we can show that the loss in (2) is convex in Δ. If $\Sigma_i \succ 0$, the unique minima for this loss function occurs at $\Delta_B^* = B_2^* - B_1^*$. So to obtain a sparse estimate of Δ_B^* using the samples of Y_i, we solve the ℓ_1-regularized optimization problem:

$$\widehat{\Delta}_B \in \arg\min_{\Delta \in \mathbb{R}^{p\times p}} \mathcal{L}(\Delta) + \lambda_n \|\Delta\|_{1,\text{off}}, \tag{3}$$

where $\lambda_n \geq 0$ and $\|\Delta\|_{1,\text{off}} = \sum_{i\neq j} |\Delta_{ij}|$ is the ℓ_1-norm applied on the off-diagonal elements of Δ. The estimate $\widehat{\Psi}_i = M_{X_i}^{-1} \widetilde{S}_i^{\frac{1}{2}} M_{X_i}^{-1}$, where $\widetilde{S}_i^{\frac{1}{2}}$ is the unique square root of the sample covariance matrix of $\widetilde{Y}_i$ and $i \in \{1, 2\}$.

In Section 4 we develop an iterative procedure to solve (3). We conclude this section by stating a result on the uniqueness of $\widehat{\Delta}_B$ in (3). Because $\|\cdot\|_1$ is convex, the combined loss function in (3) is strongly convex provided the Hessian of $\mathcal{L}(\cdot)$ is positive definite. Then, we can invoke KKT conditions for strongly convex functions to conclude that $\widehat{\Delta}_B$ is unique. However unfortunately, the Hessian matrix $H \triangleq (\widehat{\Psi}_1 \otimes \widehat{\Psi}_2 + \widehat{\Psi}_2 \otimes \widehat{\Psi}_1)/2$ is only positive semi definite. This is because $\widehat{\Psi}_i$ is positive semi definite when $p > n_i$. Hence, $\widehat{\Delta}_B$ is not unique.

Nonetheless, Lemma 1 below establishes the uniqueness of $\widehat{\Delta}_B$ by placing certain restrictions on the nullspace of the Hessian matrix. Lemma 1 is in the spirit of uniqueness results in compressed sensing. Let $\mathrm{vec}(A)$ be the mn-dimensional vector obtained by stacking the columns of $A \in \mathbb{R}^{m\times n}$ on top of each other. Let $\mathrm{vec}^{-1}(z)$, for any $z \in \mathbb{R}^{mn}$, be such that $\mathrm{vec}^{-1}(\mathrm{vec}(A)) = A$. Define the nullspace or kernel of A as $\mathrm{Ker}(A) = \{d : Ad = 0\}$.

Lemma 1. *Let H be defined as above and $d \in \mathrm{Ker}(H)$. Then $\widehat{\Delta}_B$ in (3) is unique if and only if $d^{\mathsf{T}} vec(\widehat{\Psi}_1 - \widehat{\Psi}_2) \leq 0$ and $\|vec^{-1}(d)\|_{1,off} \leq \tau$. Here, τ is given by the constrained (Lagrangian dual) form of (3) and $d \neq 0$.*

We sketch a few details of the proof here. Consider the constrained form of (3): $\arg\min_{\|\Delta\|_{1,\text{off}} \leq \tau} \mathcal{L}(\Delta, \widehat{\Psi}_1, \widehat{\Psi}_2)$. Now, rewriting the loss function in (2) in its quadratic form we have: $\mathrm{vec}(\Delta)^{\mathsf{T}} H \mathrm{vec}(\Delta)/2 - \mathrm{vec}(\Delta)^{\mathsf{T}}(\widehat{\Psi}_1 - \widehat{\Psi}_2)$. The uniqueness result then follows from [20].

4. OPTIMIZATION ALGORITHM

We solve the optimization in (3) using the alternating direction method of multipliers (ADMM) method proposed in [10]. We give high level details of this method while referring the reader to [10] for complete details.

Consider the following identity for the loss function in (2): $\mathcal{L}(\Delta, \widehat{\Psi}_1, \widehat{\Psi}_2) = (\mathcal{L}_1(\Delta))/4 + (\mathcal{L}_2(\Delta))/4$, where $\mathcal{L}_1(\Delta) = \langle \widehat{\Psi}_1 \Delta_B^*, \Delta_B^* \widehat{\Psi}_2 \rangle - 2\langle \Delta, \widehat{\Psi}_1 - \widehat{\Psi}_2 \rangle$, and similarly, $\mathcal{L}_2(\Delta) = \langle \widehat{\Psi}_2 \Delta_B^*, \Delta_B^* \widehat{\Psi}_1 \rangle - 2\langle \Delta, \widehat{\Psi}_1 - \widehat{\Psi}_2 \rangle$. The only change in these loss functions is the positioning of $\widehat{\Psi}_1$ and $\widehat{\Psi}_2$. Consider three $p \times p$ matrices Δ_1, Δ_2, and Δ_3. Then, the optimization in (3) is equivalent to

$$\arg\min_{\Delta_1 = \Delta_2 = \Delta_3} \mathcal{L}_1(\Delta_1) + \mathcal{L}_2(\Delta_2) + \lambda_n \|\Delta_3\|_1, \tag{4}$$

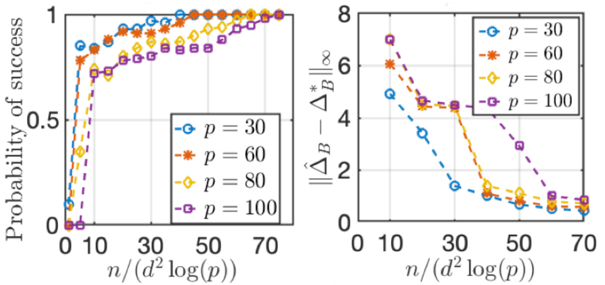
where $\widehat{\Delta}_B = \widehat{\Delta}_i$ for any $i \in \{1, 2, 3\}$. Let $\rho > 0$ be the momentum constant, and Λ_1, Λ_2, and Λ_3 be the matrix multipliers of the augmented Lagrangian of (4) (see [10] for a formula). Then, the ADMM iterates are

$$\begin{aligned}
\Delta_1^{k+1} &= G(\widehat{\Psi}_1, \widehat{\Psi}_2, 2\rho\Delta_3^k + 2\rho\Delta_2^k + \widehat{\Psi}_1 - \widehat{\Psi}_2 + \\
&\qquad 2\Lambda_1^k - 2\Lambda_3^k, 4\rho), \\
\Delta_2^{k+1} &= G(\widehat{\Psi}_2, \widehat{\Psi}_1, 2\rho\Delta_3^k + 2\rho\Delta_1^{k+1} + \widehat{\Psi}_1 - \widehat{\Psi}_2 + \\
&\qquad 2\Lambda_3^k - 2\Lambda_2^k, 4\rho), \\
\Delta_3^{k+1} &= S((\rho\Delta_1^{k+1} + \rho\Delta_2^{k+1} - \Lambda_1^k + \Lambda_2^k)/(2\rho), \lambda/2\rho), \\
\Lambda_1^{k+1} &= \Lambda_1^k + \rho(\Delta_3^{k+1} - \Delta_1^{k+1}), \\
\Lambda_2^{k+1} &= \Lambda_2^k + \rho(\Delta_2^{k+1} - \Delta_3^{k+1}), \text{ and} \\
\Lambda_3^{k+1} &= \Lambda_3^k + \rho(\Delta_1^{k+1} - \Delta_2^{k+1}).
\end{aligned}$$

The shrink function $S(\cdot)$ is defined as follows: $S(A, \lambda) = 0$ when $|A| \leq \lambda$, and $S(A, \lambda) = A - \mathrm{sign}(A)\lambda$. For any symmetric matrices P, Q, and R, and a positive γ, the function $G(\cdot)$ takes the following form: $G(P, Q, R, \gamma) \triangleq U_P\{O \circ (U_P^{\mathsf{T}} R U_Q)\} U_Q^{\mathsf{T}}$, where $\circ$ is the Hadamard product of two matrices. Further, $U_P D_P U_P^{\mathsf{T}}$ and $U_Q D_Q U_Q^{\mathsf{T}}$ are the eigendecompositions of P and Q. Finally, $O_{ij} = [D_P(j, j) D_Q(i, i) + \gamma]^{-1}$. The formula for $G(\cdot)$ in [10] is incorrect and the expression we state here is correct.

5. NUMERICAL SIMULATIONS

We illustrate the performance of our estimator on synthetic and two benchmark power systems. We consider two performance metrics: (i) the empirical probability (averaged over 100 instances) of recovering the support of Δ_B^* and (ii) the worst case error evaluated using $\|\widehat{\Delta}_B - \Delta_B^*\|_\infty$. Recall that $\|A\|_\infty = \max_{i,j} |a_{i,j}|$. In the figures below, we plot these error metrics as a function of the re-scaled sample size $n/(d^2 \log(p))$, where d is the maximum degree of Δ_B^*. This scaling is theoretically justified in [10]. We set $\lambda_n \propto \sqrt{\log(p)/n}$ and the parameter $\rho = 0.001$.



(a) support recovery rate (b) worst case error bound

Fig. 1: Estimation accuracy for synthetic networks.

Figure. 1 shows the estimation accuracy for Δ_B^* for many dimensions (p). In each case, the graph underlying Δ_B^* is a grid graph with degree $d = 4$. We can visualize this graph by letting the nodes correspond to the points in the 2D-plane with integer coordinates. For this choice of Δ_B^*, we set B_1^* to be a random, invertible symmetric matrix. We then define $B_2^* = B_1^* + \Delta_B^*$. Importantly, B_2^* and B_1^* are non-sparse. In Fig. 1(a) and (b), the accuracy improves as a function of the re-scaled sample size. But the accuracy deteriorates as the dimension (p) increases, which is expected. Notably, for fixed d, $\|\widehat{\Delta}_B - \Delta_B^*\|_\infty$ behaves approximately as $1/\sqrt{n/(\log(p))}$, which agrees with the support recovery results on sparse regression.

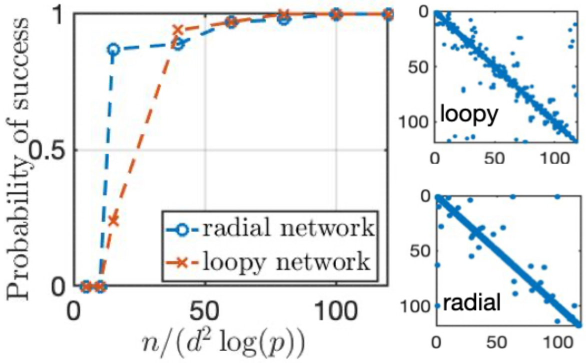


Fig. 2: Estimation accuracy for power networks.

Similar to Figure. 1, Figure. 2, shows the estimation accuracy for different choices of Δ_B^* whose underlying graphs are grids. But B_1^* and the graph underlying it are associated with an electric power network. Specifically, we consider the radial IEEE 118 bus distribution network and the loopy IEEE 118 bus transmission network [21]. As mentioned earlier, we reduced the networks by deleting a node. Hence, $p = 117$. The panels on the right visualize the sparsity patterns of the reduced networks. For both networks, the support recovery rate in the left panel increases with the re-scaled sample size. This result again confirms that the sparsity of individual networks plays no role in the estimation performance.

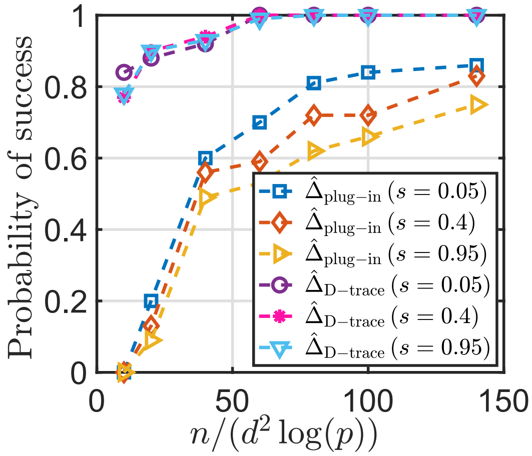


Fig. 3: Plug-in estimator vs proposed estimator.

Figure. 3 compares the support recovery rates of the proposed estimator and the naive plug-in estimator. The latter is obtained by plugging the inverse of the square root of the sample covariance matrix in (1). So, for this experiment, we assume that $n > p = 60$. We consider three matrices for B_1^*, with increasing number of zeros. We regulate the number of zeros in B_1^* using the parameter s, which is defined as the ratio of the number of non-zeros to the number of entries in the matrix. The smaller the s, the sparser is the matrix. The graph underlying Δ_B^* is grid and we let $B_2^* = B_1^* + \Delta_B^*$. As shown in Figure. 3, for every choice of s, our estimator (called D-trace in the figure), outperforms the plug-in estimator. Importantly, our estimator works well even when $n < p = 60$, where the plug-in estimator does not even exist.

6. CONCLUSION

In this paper, we consider differential network analysis for systems obeying conservation laws. For random node injections, we show that the sparsity pattern of the square root of the inverse covariance matrix of the node potential vector encodes the network structure. We exploit this property to develop an estimator that directly estimates the difference of two network Laplacian matrices using the samples of potentials. We adapt the ADMM method in [10] to numerically implement the proposed estimator. Our numerical results demonstrate the superior performance of our estimator over the standard plug-in estimator.

7. REFERENCES

[1] A. van der Schaft, "Modeling of physical network systems," *Systems & Control Letters*, vol. 101, pp. 21–27, 2017.

[2] A. Bressan, S. Čanić, M. Garavello, M. Herty, and B. Piccoli, "Flows on networks: recent results and perspectives," *EMS Surveys in Mathematical Sciences*, vol. 1, no. 1, pp. 47–111, 2014.

[3] R. Anguluri, G. Dasarathy, O. Kosut, and L. Sankar, "Grid topology identification with hidden nodes via structured norm minimization," *IEEE Control Systems Letters*, vol. 6, pp. 1244–1249, 2021.

[4] A. Rayas, R. Anguluri, and G. Dasarathy, "Learning the structure of large networked systems obeying conservation laws," in *Advances in Neural Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022, pp. 14 637–14 650.

[5] D. Deka, S. Talukdar, M. Chertkov, and M. V. Salapaka, "Graphical models in meshed distribution grids: Topology estimation, change detection & limitations," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4299–4310, 2020.

[6] T. Li, L. Werner, and S. H. Low, "Learning graphs from linear measurements: Fundamental trade-offs and applications," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 163–178, 2020.

[7] A. C. Varghese, A. Pal, and G. Dasarathy, "Transmission line parameter estimation under non-gaussian measurement noise," *IEEE Transactions on Power Systems*, 2022.

[8] A. Shojaie, "Differential network analysis: A statistical perspective," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 13, no. 2, 2021.

[9] S. Na, M. Kolar, and O. Koyejo, "Estimating differential latent variable graphical models with applications to brain connectivity," *Biometrika*, vol. 108, no. 2, pp. 425–442, 2021.

[10] H. Yuan, R. Xi, C. Chen, and M. Deng, "Differential network analysis via lasso penalized D-trace loss," *Biometrika*, vol. 104, no. 4, pp. 755–770, 2017.

[11] G. Dasarathy, P. Shah, and R. G. Baraniuk, "Sketched covariance testing: A compression-statistics tradeoff," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2017, pp. 676–680.

[12] F. Dorfler and F. Bullo, "Kron reduction of graphs with applications to electrical networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 1, pp. 150–163, 2012.

[13] R. Bhatia, "Positive definite matrices." Princeton University Press, 2009.

[14] M. Yuan and Y. Lin, "Model selection and estimation in the gaussian graphical model," *Biometrika*, vol. 94, no. 1, pp. 19–35, 2007.

[15] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical LASSO," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[16] T. Cai, W. Liu, and X. Luo, "A constrained ℓ_1 minimization approach to sparse precision matrix estimation," *Journal of the American Statistical Association*, vol. 106, no. 494, pp. 594–607, 2011.

[17] T. Zhang and H. Zou, "Sparse precision matrix estimation via lasso penalized D-trace loss," *Biometrika*, vol. 101, no. 1, pp. 103–120, 2014.

[18] Y. Wu, T. Li, X. Liu, and L. Chen, "Differential network inference via the fused D-trace loss with cross variables," *Electronic Journal of Statistics*, vol. 14, no. 1, pp. 1269–1301, 2020.

[19] H. Xudong and L. Mengmeng, "Confidence intervals for sparse precision matrix estimation via lasso penalized D-trace loss," *Communications in Statistics-Theory and Methods*, vol. 46, no. 24, pp. 12 299–12 316, 2017.

[20] Z. Dostál, "On solvability of convex noncoercive quadratic programming problems," *Journal of Optimization Theory and Applications*, vol. 143, no. 2, pp. 413–416, 2009.

[21] R. D. Zimmerman, C. E. Murillo-Sanchez, and D. Gan, "Matpower user's manual," *School of Electrical Engineering, Cornell University, Ithaca*, 2005.