Journal of Artificial Intelligence Research 80 (2024) 593-612 Submitted 02/2023; published 06/2024

Methods for Recovering Conditional Independence Graphs: A Survey

Harsh Shrivastava HSHRIVASTAVA@MICROSOFT.COM
Urszula Chajewska URSZC@MICROSOFT.COM
Microsoft Research, Redmond, USA

Abstract

Conditional Independence (CI) graphs are a type of Probabilistic Graphical Models that are primarily used to gain insights about feature relationships. Each edge represents the partial correlation between the connected features which gives information about their direct dependence. In this survey, we list different methods and study the advances in techniques developed to recover CI graphs. We cover traditional optimization methods as well as recently developed deep learning architectures along with their recommended implementations. To facilitate wider adoption, we include preliminaries that consolidate associated operations, for example techniques to obtain covariance matrix for mixed datatypes.

1. Introduction

Let's assume we have a domain of interest with a set of variables or features $\{X_1, \cdots, X_D\}$. The domain is governed by an unknown distribution P, which generates a dataset X with M samples and D features. We would like to learn an approximation to P. For all but the smallest D, fully representing the distribution P is infeasible. Computationally, such representation would be too large to fit in memory and too expensive to manipulate. Cognitively, it would be impossible to contemplate as many combinations of variable values would correspond to extremely unlikely events. Moreover, we would need huge datasets to estimate P accurately. All these problems can be (at least partially) resolved using the concept of probabilistic conditional independence.

To understand the concept of conditional independence, let's examine the difference between a pair of features X_i and X_j that are correlated and a pair of features that are directly correlated. Consider a universe with only three variables represented as an undirected graph G_{ex} = [study]—[grades]—[graduation], where we assume that each edge represents a positive correlation. We can see that if a student studies, then they will get good grades, which in turn increases their chances of graduation. We can thus conclude that study is correlated to graduation. But, if we know a student's grades, regardless of whether the student has studied, we can make conclusions about the chances of graduation. Thus, studying is not directly correlated to the graduation, whereas grades and graduation are directly correlated. Note that in the graph G_{ex} there is no edge between the variable representing study and the variable representing graduation. Assuming that the size of the parametrization of P is proportional to the number of edges in the graph, we have achieved considerable computational and cognitive gains. Taking clue from this toy example, we can envision that such analysis can be very useful to obtain valuable insights from the input data as well as leverage the natural interpretability provided by the graph representation.

Formally, we define the Conditional Independence (CI) graph of a set of random variables $X_i's$ to be the undirected graph $G = (D, E)$ where $D = \{1, 2, \ldots, d\}$ are nodes and (i, j) is not in the edge set if and only if $X_i \perp\!\!\!\perp X_j | \mathbf{X}_{D\setminus i,j}$, where $\mathbf{X}_{D\setminus i,j}$ denotes the collection of all of the random variables except for X_i and X_j. We assume that the set of conditional independence properties encoded in the graph reflects independence properties of the distribution used to generate input data. More formally, let's define $\mathcal{I}(P)$ to be the set of independence assertions of the form $(X_i \perp\!\!\!\perp X_j | X_k)$ that hold in the data generating distribution P. If P satisfies all independence assertions encoded by G, that is, $\mathcal{I}(G) \subseteq \mathcal{I}(P)$, we say that G is an I-map (independence map) of P. Since the complete graph is an I-map for any distribution, we are typically interested in a minimal I-map, that is, an I-map graph such that a removal of a single edge will render it not an I-map. If $\mathcal{I}(G) = \mathcal{I}(P)$, the graph G is a perfect map of P. However, it is important to keep in mind that not every distribution has a perfect map.

The second defining property of the Conditional Independence graph is that it is parameterized by edge weights that represent partial correlations between the features. In that case the edge weights will range between $e_w \sim (-1, 1)$. In the rest of the paper, we will use the term *Conditional Independence graphs* to denote graphs that are minimal I-maps of the underlying distribution and use such a parameterization.

CI graphs are primarily used to gain insights about feature relationships to help with decision making. In some cases, they are also used to study the evolving feature relationships with time. The focus of this paper is to review different methods and recent techniques developed to recover CI graphs. We will start by giving a brief overview of algorithms that recover different types of graphs.

1.1 Graph Recovery Approaches

The field of graph recovery approaches has grown considerably in recent years. Fig. 1 attempts to list popular formulations of graph representations and representative algorithms to recover the same. The algorithms that recover Conditional Independence graphs parameterized to represent partial correlations are the focus of this survey and are discussed in Sec. 2. For the sake of better understanding of this space, we will briefly describe approaches that recover graphs, edges of which not necessarily represent partial correlations between nodes.

Regression Based Methods. This line of research follows the idea of fitting a regression between features of the input data $\mathbf{X}$ to find dependencies among them. This approach is particularly popular for recovering Gene Regulatory Networks (GRN) where the input gene expression data have D genes and M samples, $X \in \mathbb{R}^{M\times D}$. Generally, the objective function used for graph recovery is a variant of the regression between the expression value of each gene as a function of the other genes (or alternatively transcription factors) and some random noise $X_d = f_d\left(X_{D\setminus d}\right) + \epsilon$, $\forall d \in D$. Usually, a sparsity constraint is also associated with the regression function to identify the top influencing genes for every gene. Many methods have been developed specifically for GRN recovery with varied choice of the regression function f_d. TIGRESS (Haury, Mordelet, Vera-Licona, & Vert, 2012) modeled f_d as linear, GENIE3 (Van Anh Huynh-Thu, Wehenkel, & Geurts, 2010) took each f_g to be a random forest while GRNBoost2 (Moerman, Aibar Santos, Bravo

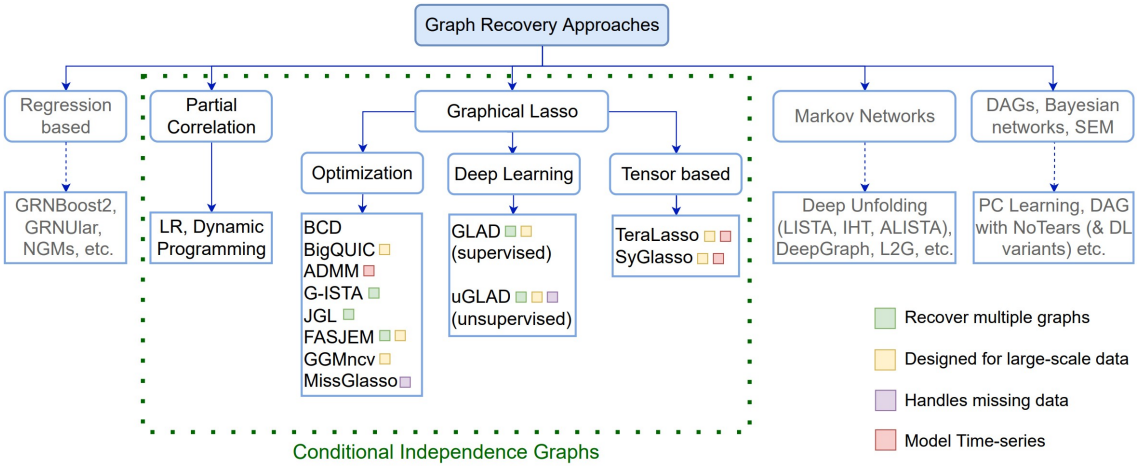


Figure 1: **Graph recovery approaches.** Methods used to recover Conditional Independence graphs are the focus of this survey. The recovered CI graph shows partial correlations between the feature nodes. The algorithms (leaf nodes) listed here are representative of the sub-category and the list is not exhaustive.

Gonzalez-Blas, Simm, Moreau, Aerts, & Aerts, 2019) used gradient boosting technique. More recently, neural network based representations like GRNUlar (Shrivastava, Zhang, Aluru, & Song, 2020; Shrivastava, Zhang, Song, & Aluru, 2022) were developed. Recently proposed Neural Graphical Models (NGMs) (Shrivastava & Chajewska, 2023b), Neural Graph Revealers (NGRs) (Shrivastava & Chajewska, 2023a; Shrivastava, 2023) used neural networks as a multitask learning framework to fit regressions and recover graph for generic input datatypes.

Markov Networks. Markov networks are probabilistic graphical models defined on undirected graphs that follow the Markov properties (Koller & Friedman, 2009). While Markov networks follow the conditional independence properties (pairwise, local and global), we made the distinction from the CI graphs based on the interpretation of edge connections. There are traditional constraint-based and score-based structure learning methods to learn Markov networks (Koller & Friedman, 2009). However, these methods suffer from combinatorial explosion of computation requirements and often simplifying approximations are made. Recently, (Belilovsky, Kastner, Varoquaux, & Blaschko, 2017) designed a supervised deep learning architecture to learn mapping from the samples to a graph, called DeepGraph. Their model had a considerable number of learning parameters, while the performance showed limited success. On the other hand, the *deep unfolding* or *unrolled algorithm* methodology for estimating a sparse vector like Iterative Shrinkage Thresholding Algorithm (ISTA) (Gregor & LeCun, 2010), ALISTA (Liu & Chen, 2019) and others (Sun, Li, Xu, et al., 2016; Chen, Liu, Wang, & Yin, 2018; Chen, Chen, Chen, Wang, Heaton, Liu, & Yin, 2022) that were primarily developed for other applications (e.g., compressed sensing) have been adopted to recover Markov networks. (Pu, Cao, Zhang, Dong, & Chen, 2021) proposed a deep unfolding approach, named L2G, to learn graph topologies. Their framework can also unroll a primal-dual splitting algorithm (Komodakis & Pesquet, 2015; Kalofolias, 2016) into a neural network for their model.

Directed Graphs. This is a very active area of research with lots of new methods being developed at a rapid pace. Directed Acyclic Graphs, Bayesian networks, structural equation models are the prominent types of directed graphs of interest. PC-learning algorithm was one of the first techniques developed to learn Bayesian Networks (Spirtes & Meek, 1995), followed by a suite of score-based and constraint-based learning algorithms and their parallel variants (Heckerman, Geiger, & Chickering, 1995; Koller & Friedman, 2009). (Zheng, Aragam, Ravikumar, & Xing, 2018) introduced "DAG with NOTEARS" method which converted the combinatorial optimization problem of DAG learning to a continuous one. This led to development of many follow up works including some deep learning methods like (Yu, Chen, Gao, & Yu, 2019; Zhang, Jiang, Cui, Garnett, & Chen, 2019; Zheng, Dan, Aragam, Ravikumar, & Xing, 2020; Pamfil, Sriwattanaworachai, Desai, Pilgerstorfer, Georgatzis, Beaumont, & Aragam, 2020) to name a few. (Heinze-Deml, Maathuis, & Meinshausen, 2018) provide a review of the causal structure learning methods and structural equation models.

In general, CI graphs are preferred choice of learning underlying graphical representations as they provide a good balance between representation capability (multivariate Gaussian distribution), fast and scalable methods for recovery, interpretability and easy probabilistic inference and querying of the resulting graphical model.

2. Recovering Conditional Independence Graphs

We define the scope of Conditional Independence (CI) graphs as undirected probabilistic graphical models that show partial correlations between features. In early frameworks, the Conditional Independence graphs were restricted to continuous variables only, which severely limited their applicability to real-world data. Before we deep dive into the algorithms that recover CI graphs, covered by the green envelope in Fig. 1, we provide a primer on handling input data with mixed datatypes. Encountering a mix of numerical (real, ordinal) and categorical variables in the input is very common. Since many of the CI graph recovery models take covariance matrix $\text{cov}(X) \in \mathbb{R}^{D\times D}$ as input, one way to accommodate discrete data is to calculate the covariance matrix with the categorical variables included.

2.1 Covariance Matrix for Mixed Datatypes

We describe ways to calculate the covariance matrix for inputs $\mathbf{X}$ with M samples and D features consisting of variables of numerical and categorical types. The value of each entry of $\text{cov}(\mathbf{X}) \in \mathbb{R}^{D\times D}$ depends on the type of interacting features, say X_i, X_j, and will be one of the following:

(I) *numerical-numerical correlation.* The obvious choice is the Pearson correlation coefficient, with range between $[-1, 1]$, defined as $\rho_{X_i,X_j} = \frac{\mathbb{E}\left[(X_i-\mu_{X_i})(X_j-\mu_{X_j})\right]}{\sigma_{X_i}\sigma_{X_j}}$, where σ_{X_i} denotes standard deviation and μ_{X_i} is the mean of the feature X_i and similarly for X_j. The Pearson correlation assumes that the variables are linearly related and can be quite sensitive to outliers. To capture non-linear relationships, ordinal association (or rank correlation) based metrics like Spearman's correlation coefficient, Kendall's τ (tau), Goodman and Kruskal's gamma or Somers' D can be leveraged.

(II) *categorical-categorical association.* The most common measure of association between two categorical variables is Cramér's V statistic along with the bias correction. The association value is in the range of $[0, 1]$, where 0 means no association and 1 is full association between categorical features. Consider two categorical features C^1 and C^2 with sample size M and have $i = \{1, \cdots, p\}$, $j = \{1, \cdots, q\}$ be the possible categories for C^1 and C^2, respectively. Let m_{ij} denote the number of times the values (C_i^1, C_j^2) occur. The Cramér's V statistic is defined as $V = \sqrt{\frac{\chi^2/m}{\min(p-1, q-1)}}$, where $\chi^2 = \sum_{ij} \frac{\left(m_{ij} - \frac{m_{i*}m_{*j}}{m}\right)^2}{\frac{m_{i*}m_{*j}}{m}}$ is the chi-square statistic, $m_{i*} = \sum_j m_{ij}$ and $m_{*j} = \sum_i m_{ij}$. It has been observed that Cramér's V statistic tends to overestimate the association strength. To address this issue, a bias correction modification was introduced as $\tilde{V} = \sqrt{\frac{\tilde{\varphi}^2}{\min(\tilde{p}-1, \tilde{q}-1)}}$, where $\tilde{\varphi}^2 = \max\left(0, \chi^2/m - \frac{(p-1)(q-1)}{(m-1)}\right)$, $\tilde{p} = p - \frac{(p-1)^2}{(m-1)}$ and $\tilde{q} = q - \frac{(q-1)^2}{(m-1)}$. Some more interesting approaches using Gini index and word2vec representation are discussed in (Niitsuma & Lee, 2016) and can also be utilized.

(III) *categorical-numerical correlation.* There are multiple options available like the correlation ratio (range is $[0, 1]$) , point biserial correlation (range is $[-1, 1]$) , Kruskal-Wallis test by ranks (or H test). Each of these methods have their advantages and drawbacks that should be considered while selecting the metric. Yet another way can be to bin the numerical variable and convert it to a categorical variable. Then Cramér's V can be used to calculate the correlation.

Refer to (Sheskin, 2003) for details about the aforementioned statistical techniques.

2.2 Methods

Recovery of Conditional Independence graphs is a topic of wide research interest. There are two popular formulations to recover CI graphs: the first one directly determines partial correlation values, the second derives them by utilizing matrix inversion approaches. Based on the formulation chosen, many optimization algorithms have been developed with each having their own capabilities and limitations.

2.2.1 Direct Calculation of Partial Correlation Values

The definition of the partial correlation between two features X_i and X_j given a set of $D-2$ controlling variables $\mathbf{X}_{D\setminus i,j}$, written $\rho_{X_i,X_j}.\mathbf{X}_{D\setminus i,j}$, is the correlation between the residuals e_{X_i} and e_{X_j} after fitting a linear regression of X_i with $\mathbf{X}_{D\setminus i,j}$ and of X_j with $\mathbf{X}_{D\setminus i,j}$, respectively. Poplular approaches used to obtain the partial correlation values directly are discussed below.

Linear Regression. The regressions for the partial correlation calculations are formulated using linear functions. Let vectors $\{\mathbf{w}_i, \mathbf{w}_j\} \in \mathbb{R}^{D-1}$ and $\mathbf{X}_{D\setminus i,j}$ denote the vector of the other features augmented by 1 to account for bias. Then the regression over the M samples will be

$$\mathbf{w}_k = \arg\min_{\mathbf{w}} \sum_{m=1}^{M} X_k^m - \langle \mathbf{w}, \mathbf{X}_{D\setminus i,j}^m \rangle, \quad \text{where k=\{i, j\}.} \tag{1}$$

We then calculate the residuals for each individual sample as $e_{X_k}^m = X_k^m - \langle \mathbf{w}, \mathbf{X}_{D\setminus i,j}^m \rangle$ where k=\{i, j\}. The partial correlation between X_i, X_j, which is the $\{i, j\}$ entry of the matrix

$\mathrm{P} \in \mathbb{R}^{D \times D}$ is calculated as the correlation between the residuals,

$$\rho_{X_i,X_j}.\mathbf{X}_{D\setminus i,j} = \frac{M \sum_{m=1}^{M} e_{X_i}^m e_{X_j}^m}{\sqrt{M \sum_{m=1}^{M} (e_{X_i}^m)^2} \sqrt{M \sum_{m=1}^{M} (e_{X_j}^m)^2}} \tag{2}$$

Recursive Formulation. The linear formulation is computationally expensive, which is its major drawback. The recursive formulation uses dynamic programming based algorithm to recursively calculate the following partial correlation expression, for any $X_k \in \mathbf{X}_{D\setminus i,j}$,

$$\rho_{X_i,X_j}.\mathbf{X}_{D\setminus i,j} = \frac{\rho_{X_i,X_j}.\mathbf{X}_{D\setminus i,j,k} - \rho_{X_i,X_k}.\mathbf{X}_{D\setminus i,j,k} \times \rho_{X_k,X_j}.\mathbf{X}_{D\setminus i,j,k}}{\sqrt{1 - \rho^2_{X_i,X_k}.\mathbf{X}_{D\setminus i,j,k}} \sqrt{1 - \rho^2_{X_k,X_j}.\mathbf{X}_{D\setminus i,j,k}}} \tag{3}$$

Refer to (Baba, Shibata, & Sibuya, 2004) for a thorough treatment about partial correlations as measures of conditional independence between variables.

2.2.2 Graphical Lasso & Variants

Given M observations of a D-dimensional multivariate Gaussian random variable $X = [X_1, \ldots, X_D]^\top$, the sparse graph recovery problem aims to estimate its covariance matrix Σ^* and precision matrix $\Theta^* = (\Sigma^*)^{-1}$. The ij-th component of Θ^* is zero if and only if X_i and X_j are conditionally independent given the other variables $\{X_k\}_{k \neq i,j}$. The general form of the graphical lasso optimization to estimate Θ^* is the minimization of the log-likelihood of a multivariate Gaussian with regularization as

$$\widehat{\Theta} = \arg\min_{\Theta \in \mathcal{S}^D_{++}} \quad -\log(\det \Theta) + \mathrm{tr}(\widehat{\Sigma}\Theta) + \mathrm{Reg}(\Theta_{\text{off}}), \tag{4}$$

where $\widehat{\Sigma}$ is the empirical covariance matrix based on M samples, $\mathcal{S}^D_{++}$ is the space of $D \times D$ symmetric positive definite matrices and $\mathrm{Reg}(\Theta_{\text{off}})$ is the regularization term for the off-diagonal elements. Once the precision matrix is obtained, the corresponding partial correlation matrix entries can be calculated as $\rho_{X_i,X_j}.\mathbf{X}_{D\setminus i,j} = -\frac{\widehat{\Theta}_{i,j}}{\sqrt{\widehat{\Theta}_{i,i}\widehat{\Theta}_{j,j}}}$. Several algorithms have been developed to optimize the sparse precision matrix estimation problem in Eq. 4 which primarily differ in the choice of regularization and optimization procedure.

Block Coordinate Descent (BCD). (Banerjee, Ghaoui, & d'Aspremont, 2008) formulated the graphical lasso problem of approximating precision matrix as the ℓ_1-regularized maximum likelihood estimation

$$\widehat{\Theta} = \arg\min_{\Theta \in \mathcal{S}^D_{++}} \quad -\log(\det \Theta) + \mathrm{tr}(\widehat{\Sigma}\Theta) + \lambda \left\|\Theta\right\|_{1,\text{off}}, \tag{5}$$

where $\|\Theta\|_{1,\text{off}} = \sum_{i \neq j} |\Theta_{ij}|$ is the off-diagonal ℓ_1 regularizer with regularization parameter λ. Block-coordinate descent methods, for example (Friedman, Hastie, & Tibshirani, 2008), update each row (and the corresponding column) of the precision matrix iteratively by solving a sequence of lasso problems. A variant of this algorithm is the popular `GraphicalLasso` function implementation of python's scikit-learn package (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot, & Duchesnay, 2011). It is very efficient for large scale problems

involving thousands of variables. This estimator is sensible even for non-Gaussian input data, since it is minimizing an ℓ_1-penalized log-determinant Bregman divergence (Ravikumar, Wainwright, Raskutti, & Yu, 2011). Different variants of solvers have been proposed to handle large scale data, some of the prominent ones being `BigQUIC, QUIC, SQUIC` by (Hsieh, Sustik, Dhillon, Ravikumar, & Poldrack, 2013; Hsieh, Sustik, Dhillon, Ravikumar, et al., 2014; Bollhofer, Eftekhari, Scheidegger, & Schenk, 2019) respectively, can handle up to 1 million random variables.

G-ISTA. The Graphical Iterative Shrinkage Thresholding Algorithm was proposed by (Rolfs, Rajaratnam, Guillot, Wong, & Maleki, 2012). This method uses proximal gradient descent based approach to perform ℓ_1-regularized inverse covariance matrix estimation specified in Eq. 5. The basic idea is to separate the continuously differentiable, convex function (first two terms of Eq. 5) and the regularization term which is convex but not necessarily smooth. Then the standard updates of general iterative shrinkage thresholding algorithm (ISTA) (Beck & Teboulle, 2009) can be applied. The G-ISTA algorithm comes with nice theoretical and stability properties. Similarly, Alternating Direction Method of Multipliers (ADMM) (Boyd, Parikh, Chu, Peleato, Eckstein, et al., 2011) can be used to optimize the ℓ_1 regularized objective.

Graphical Non-Convex Optimization. (Sun, Tan, Liu, & Zhang, 2018) proposed graphical non-convex optimization for optimal estimation in Gaussian graphical models. They consider the optimization objective of Eq. 4 with the $\text{Reg}(\Theta_{\text{off}}) = \sum_{i \neq j} \lambda(\Theta_{i,j})$, where $\lambda(\cdot)$ is a non-convex penalty. This non-convex optimization is then approximated by a sequence of adaptive convex programs. The experiments demonstrate improvements over previous methods and thus advocate for development of methods that account for non-convex penalties. They note that their algorithm is an adaptive version of the SPICE algorithm by (Rothman, Bickel, Levina, Zhu, et al., 2008). Another recent work by (Zhang, Fattahi, & Sojoudi, 2018) builds up on prior work that shows that the graphical lasso estimator can be retrieved by soft-thresholding the sample covariance matrix and solving a maximum determinant matrix completion (MDMC) problem. They proposed a Newton-CG algorithm to efficiently solve the MDMC problem. The authors claim it to be highly efficient for **large scale** data. We note that there are many works which model sparsity constraints differently or provide surrogate objective functions. We list a few prominent ones here to help the readers get an overview (Loh & Wainwright, 2011; Yang, Lozano, & Ravikumar, 2014; Sojoudi, 2016; Zhang et al., 2018). A collection of methods, `GGMncv`, for Gaussian Graphical models with non-convex regularization can be found in (Williams, 2020).

`GLAD`. (Shrivastava et al., 2020; Shrivastava, 2020) proposed a supervised deep learning based model, presented in Fig. 2, to recover sparse graphs based on the graphical lasso objective. They built up on the theoretically proven advantages of having non-convex penalty in the graphical lasso objective. They also proved that it is beneficial to have adaptive sequence of penalty hyperparameters for the regularization term as it leads to faster convergence. Specifically, they applied the Alternating Minimization (AM) algorithm to the objective in Eq. 5 and unrolled the AM algorithm to certain number of iterations, also known as 'deep unfolding' technique. Then, the hyperparamters were parameterized using small neural networks for doing operations like entry-wise thresholding of the precision matrix. Their deep model `GLAD` has significantly fewer number of learnable parameters

along with being fully interpretable as one can inspect the recovered graph at any point of optimization. `GLAD` was successful in avoiding any post-processing requirements to maintain the symmetric and SPD properties of the recovered precision matrix. The training of `GLAD` model was done using supervision and the hope was that the model can generalize over that underlying distribution of graphs. They were first to demonstrate that learning can help improve sample complexity.

`uGLAD`. Proposed by (Shrivastava, Chajewska, Abraham, & Chen, 2022a, 2022b), `uGLAD` is an unsupervised deep model that circumvents the need of supervision which was the key bottleneck of the `GLAD` model. They changed the optimization problem by introducing the *glasso* loss function and incorporating the regularization in the deep model architecture (defined by `GLAD`) itself which is implictly learned during optimization. The input of `uGLAD` is the empirical covariance matrix; it requires no ground-truth information to optimize. It can also perform multitask learning by optimizing **multiple CI graphs** at once. Some other prominent methods that recover multiple graphs include `JGL` (Danaher, Wang, & Witten, 2014) and `FASJEM` (Wang, Gao, & Qi, 2017). Furthermore, `uGLAD` leverages this ability to handle **missing data** by introducing a consensus strategy and thus have more robust performance. A different approach to handle missing data have been previously explored in `MissGLasso` (Stadler & Buhlmann, 2012), (Loh & Wainwright, 2011) among other methods.

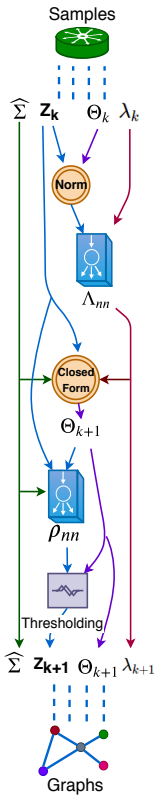


Figure 2: The recurrent unit `GLADcell`. (Taken from (Shrivastava et al., 2020))

`Tensor Graphical Lasso`. (Greenewald, Zhou, & Hero III, 2019) proposed `TeraLasso` that extends the graphical lasso problem to higher-order tensors. They introduced a multiway tensor generalization of the bi-graphical lasso which uses a two-way sparse Kronecker sum multivariate normal model for the precision matrix to model parsimoniously conditional dependence relationships of matrix variate data based on the Cartesian product of graphs. The Sylvester Graphical Lasso or `SyGlasso` model by (Wang, Jang, & Hero, 2020) complements `TeraLasso` by providing an alternative Kronecker sum model that is generative and interpretable. These approaches are typically very helpful in modeling spatio-temporal data.

2.3 Properties of Recovery Algorithms

While describing individual algorithms, we touched upon some of the desirable properties that such methods should have. Let's look at some of them in detail.

Recovering multiple CI graphs at once. Most of the work in learning CI graphs has focused on estimating a single model. In recent years, however, the framework was extended to jointly fitting a collection of such models based on data that share the same variables, with dependency structure varying with some external category. For example, when fitting a model for data obtained from anaerobic digesters, we may want to learn separate models for digesters operating at different temperatures. Some of the feature relationships will turn out to be independent of the temperature setting and stay the same between the graphs, but

Methods	Implementation	MG	LS	MI	TS	Paper
BCD	Scikit-learn package					(Pedregosa et al., 2011)
QUIC,BigQUIC	Software link		✓			(Hsieh et al., 2013, 2014)
ADMM	https://github.com/tpetaja1/tvgl				✓	(Hallac, Park, Boyd, & Leskovec, 2017)
G-ISTA	Python package	✓				(Rolfs et al., 2012)
JGL	R package	✓				(Danaher et al., 2014)
FASJEM	https://github.com/QData/FASJEM	✓	✓			(Wang et al., 2017)
GGMncv	R package		✓			(Williams, 2020)
Newton-CG(MDMC)	Matlab package		✓			(Zhang et al., 2018)
MissGlasso	R package			✓		(Stadler & Buhlmann, 2012)
TeraLasso	https://github.com/kgreenewald/teralasso		✓		✓	(Greenewald et al., 2019)
SyGlasso	https://github.com/ywa136/syglasso		✓		✓	(Wang et al., 2020)
GLAD	https://github.com/Harshs27/GLAD	✓	✓		✓	(Shrivastava et al., 2020)
uGLAD	https://github.com/Harshs27/uGLAD	✓	✓	✓	✓	(Shrivastava et al., 2022b)

Table 1: Conditional Independence graph recovery methods with their implementation links. Additional information about their ability to recover multiple graphs (MG), handle large scale data (LS), handle missing values in data (MI) and to model time-series (TS) are mentioned alongside.

some will differ between models. Recovering multiple graphs is also useful while considering reconstruction of Gene Regulatory Networks from microarray expression data coming from a cancerous tissue and a benign one. This is because estimating separate graphical models for cancerous and benign tissues does not exploit the similarity between the true underlying distributions or graphical representations. If we just estimate a single graphical for both the tissue types, then we miss the important differences that sets apart cancerous cells. Additionally, optimizing for multiple graphs together lets us take advantage of the larger sample size. It also makes the process more robust in case of anomalous data. Methods for recovering multiple CI graphs, like `JGL` (Danaher et al., 2014), `FASJEM` (Wang et al., 2017) introduce an additional regularization term or a convex penalty term that connects the multiple optimization tasks. The design of such penalty terms is user dependent and varies based on the task. Whereas, newer deep learning methods like `GLAD` (Shrivastava et al., 2020; Shrivastava, 2020), and `uGLAD` (Shrivastava et al., 2022a, 2022b) update their model parameters themselves to optimize the joint graphical lasso objective function. The newer deep model design inherently accounts for the explicit user-defined penalties by the earlier methods and thus can be potentially more robust.

Handling mixed data types. As explained in Sec. 2.1, methods that take the empirical covariance matrix as input, can be extended to handle mixed data types, which makes them much more widely applicable to real-world problems.

Handling missing data. Missing data are ubiquitous in data modeling. Sensors fail, lab errors happen, and people refuse to answer some of the questions in surveys. Moreover, in most real-world situations, data is not missing completely at random (MCAR). None of the methods for dealing with missing data is completely satisfactory: dropping samples with missing data can introduce bias and reduce the size of the dataset; even state-of-the-art imputation methods may have undesirable properties (Chen, Tan, Chajewska, Rudin, & Caruana, 2023). It is preferable that the algorithms handle missing data natively, rather than rely on pre-processing steps. Of the algorithms discussed, only `MissGLasso` (Stadler & Buhlmann, 2012) and `uGLAD` (Shrivastava et al., 2022a, 2022b) do that.

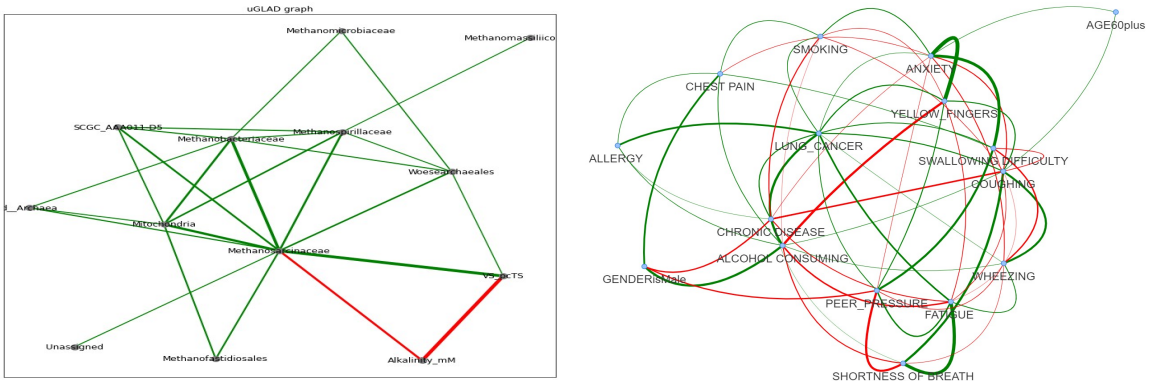


Figure 3: [left] `uGLAD` graph for archaea at family level in a collection of wastewater processing digesters. Edge color indicates the sign of the correlation: green - positive, red - negative, edge weight corresponds to correlation's strength (taken from (Shrivastava et al., 2022b)). [right] CI graphs from `uGLAD` model used to analyse a lung cancer data from (Kaggle, 2022).

Scalability. Most of the algorithms presented in Sec. 2 can scale to a large number of features and samples. However, for some methods in their base form, for instance `G-ISTA`, it is not clear how to run them for a large number of features as they have not been evaluated extensively on such high-dimensional settings in their original works.

Adapatability to time-series data. The tensor-valued Gaussian distributions opened up an interesting possibility of modeling time-series data, refer to the `TeraLasso` method. A recent work `tGLAD` by (Imani & Shrivastava, 2023) made use of CI graphs for doing multivariate time series segmentation. Specifically, they utilized the capability of `uGLAD` to do multitask learning and handling of missing data to find pattern similarities in a multivariate time-series. These methods provide an additional component of interpretability to the analysis in terms of the CI graphs that are recovered, which makes them an attractive tool for time-series data handling.

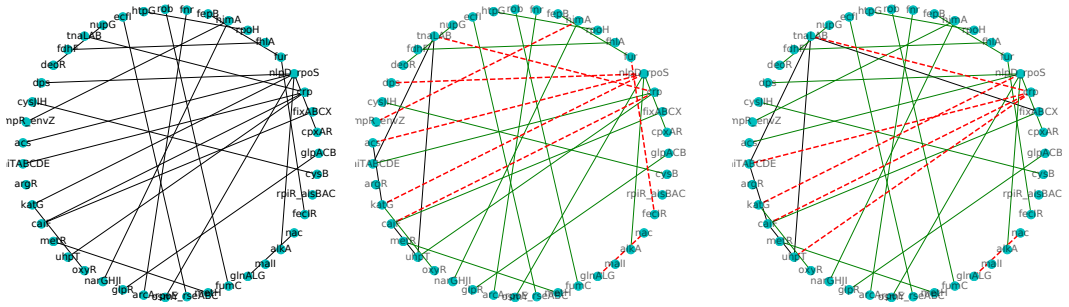
Table 1 lists some of the prominent methods for CI graph recovery along with their recommended implementations. This compilation will help the readers choose the right models for their applications. Now that we have discussed some of the popular approaches to recover CI graphs, for the sake of completeness and wider adoption, we list some of the potential applications that the CI graphs have been applied to in the past as well as hint at several unexplored opportunities to leverage them.

3. Applications of CI Graphs

CI graph recovery algorithms have been successfully applied in a variety of domains. The list below includes both actual domains and potential applications where the CI graph recovery algorithms can be applied with a potential of improvement over the current state-of-the-art.

Life Sciences. CI graphs were successfully used to study the microbials inside an anaerobic digester and to help choose system design parameters of the digester, see Fig. 3 (Shrivastava et al., 2022b). Recovering GRNs from the corresponding microarray expression data and possibly extending to ensemble methods (Guo, Jiang, Chen, & Guo, 2016; Aluru,

Shrivastava, Chockalingam, Shivakumar, & Aluru, 2022) can also be interesting to explore using CI graph recovery methods. Recovered GRNs using `GLAD` are shown in Fig. 4.



(a) True graph (b) **M=10**, fdr=0.613, tpr=0.913, fpr=0.114 (c) **M=100**, fdr=0.236, tpr=0.986, fpr=0.024

Figure 4: Recovered graph structures for a sub-network of the *E. coli* consisting of 43 genes and 30 interactions with increasing number of samples. `GLAD` was trained using ground truth from a synthetic gene expression data simulator. Increasing the number of samples reduces the FDR by discovering more true edges. Notation: TPR: True Positive Rate, FPR: False Positive Rate, FDR: False Discovery Rate. (taken from (Shrivastava et al., 2020)).

Medical Informatics. Graphical models have been widely used for making informed medical decisions. For instance, the PathFinder project by (Heckerman, Horvitz, & Nathwani, 1992; Heckerman & Nathwani, 1992) is a prime example of an early system using Bayesian networks for assisting medical professionals in making critical decisions. The conditional dependencies for this project were procured by consulting the doctors which can be quite time consuming and not easily scalable. Since then, many other medical expert systems based on Bayesian networks (with parameters usually learned from data) have been developed, see (McLachlan, Dube, Hitman, Fenton, & Kyrimi, 2020) for a survey. CI graphs can be used as an alternative to approximate a distribution over medical variables. For instance, Fig. 3 shows a CI graph for studying patients' data for Lung cancer prediction from a Kaggle dataset. Similarly, CI graphs are the basis of systems for discovering dependencies between important body vitals of ICU patients (Bhattacharya, Rajan, & Shrivastava, 2019; Shrivastava, Huddar, Bhattacharya, & Rajan, 2021). Another instance is shown in Fig. 5, where the authors used a CI graph recovery algorithm to analyse feature connections to study infant mortality in the US.

Protein Structure Recovery. Deep models for CI graph recovery like `uGLAD` can be substituted for predicting the contact matrix from the input correlation matrix between the amino acid sequences. For instance, Protein Sparse Inverse COVaraince or PSICOV (Jones, Buchan, Cozzetto, & Pontil, 2012), which uses graphical lasso based approach to predict the contact matrix in order to eventually predict the 3D protein structure could leverage these recently developed CI graph recovery deep models. Learnable parameters of these models can also account for the ground truth data, if available. DeepContact model (Liu, Palmedo, Ye, Berger, & Peng, 2018) uses a Convolutional Neural Network based architecture to do a matrix inversion operation for predicting contact map from the co-evolution map obtained

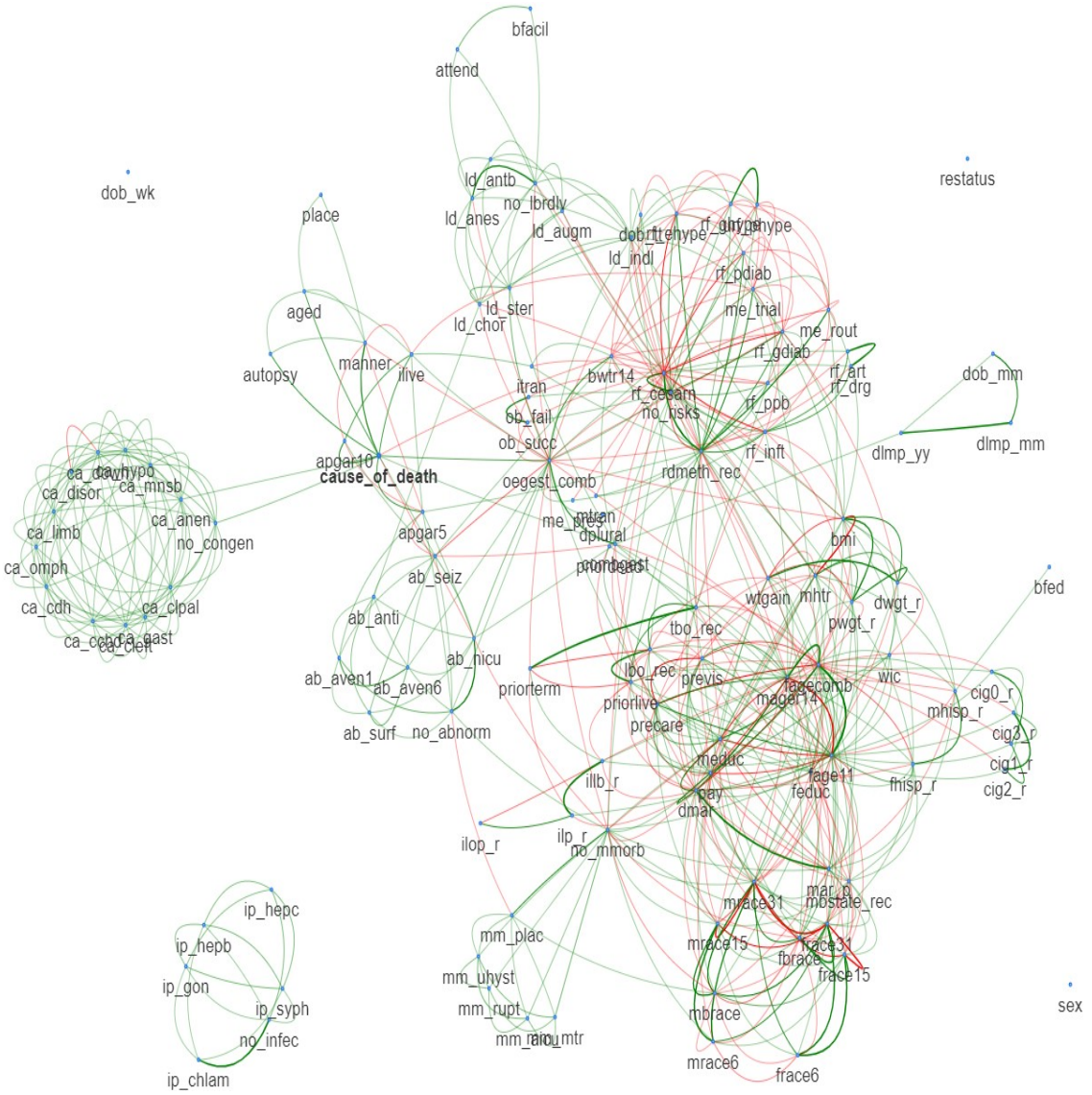


Figure 5: The CI graph recovered by `uGLAD` for the Infant Mortality 2015 data from CDC (United States Department of Health and Human Services, Division of Vital Statistics (DVS), 2015) (taken from (Shrivastava & Chajewska, 2023b)).

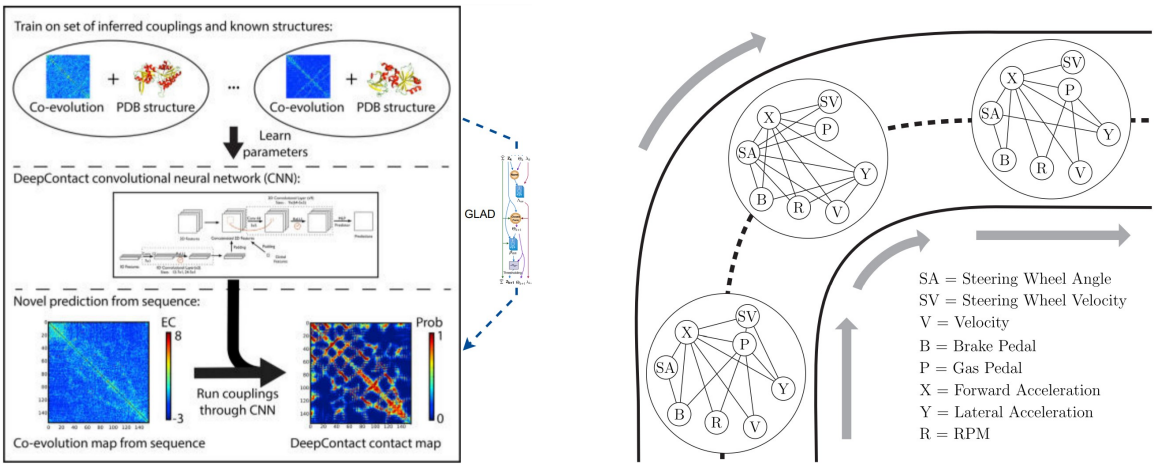


Figure 6: [left] Potential use of `GLAD` or `uGLAD` for contact map prediction to recover protein structure. [right] A dynamic network inference framework showing three snapshots of the automobile sensor network measuring eight sensors, taken (1) before, (2) during, and (3) after a standard right turn. CI graphs were recovered by running a scalable message passing algorithm based on the Alternating Direction Method of Multipliers (ADMM) to study automobile sensor network (taken from (Hallac et al., 2017)).

from protein sequences. Deep models for CI graph recovery can potentially augment (or even replace) the CNNs for improved predictions, refer to the left side of Fig. 6.

Class Imbalance Handling. Correlations discovered by the CI graphs can be helpful in narrowing down important feature clusters for identifying key features. This will in-turn improve performance in cases where there is little data or imbalanced data (more data points for one class than another). Sampling from these graphs can balance out the data, similar to the SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) procedure. CI graphs can act as preprocessing steps for some of the methods for class imbalance handling like (Rahman & Davis, 2013; Shrivastava, Huddar, Bhattacharya, & Rajan, 2015; Bhattacharya, Rajan, & Shrivastava, 2017).

Finance. CI graphs are useful for finding correlations between stocks to see how companies compare (Hallac et al., 2017).

Video Sequence Predictions. Deep models for CI graph recovery, `GLAD` or `uGLAD`, can be integrated into a pipeline for latest models used for generating unseen future video frames (Denton & Fergus, 2018; Shrivastava & Shrivastava, 2020). Specifically, in conjunction with the generative deep models, the CI graph recovery model parameters can be learned to narrow down potential future viable frames from the generated ones.

Gaussian Processes and Time Series Problems. An interesting use case by (Chatrabgoun, Soltanian, Mahjub, & Bahreini, 2021) combines graphical lasso with Gaussian processes for learning gene regulatory networks. Similarly, in a recent work on including negative data points for Gaussian processes (Shrivastava, Shrivastava, & Shrivastava, 2020), CI graphs can be used for narrowing down the relevant features for performing GP regression and for time-series modeling (Jung, Hannak, & Goertz, 2015). An example using an automobile system is shown in Fig. 6 on the right (Hallac et al., 2017). (Greenewald et al., 2019) used tensor based formulation of graphical lasso to analyse spatio-temporal data of

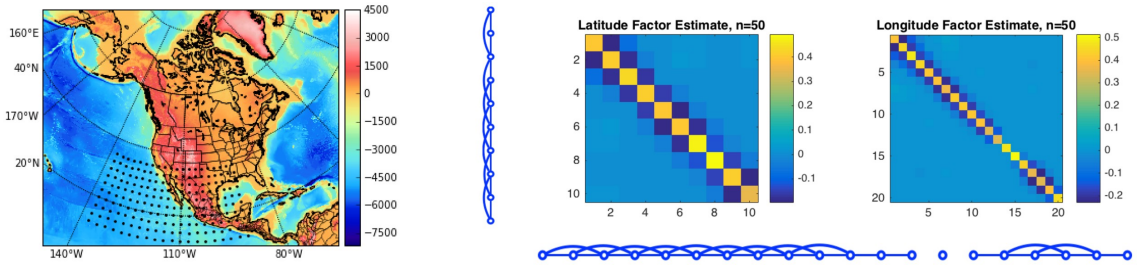
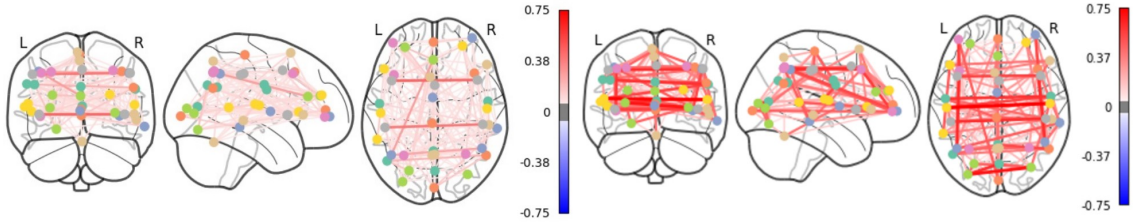


Figure 7: `TeraLasso`, a tensor based method, used for analysis of a time series data of daily-average wind speeds of the western grid of North America. [left] Rectangular 10×20 latitude-longitude grids of windspeed locations shown as black dots. Elevation colormap shown in meters. [right] Graphical representation of latitude (left) and longitude factors (bottom) with the corresponding precision estimates. Observe the decorrelation (longitude factor entries connecting nodes 1-13 to nodes 14-20 are essentially zero) in the Western longitudinal factor, corresponding to the high-elevation line of the Rocky Mountains. (taken from (Greenewald et al., 2019)).



(a) Autistic group (b) Control group

Figure 8: The connectivity of the 39 regions in the brain estimated by using 35 subjects. The CI graph was recovered by the L2G algorithm, which is a deep unfolding approach to learn graph topologies, refer to (Pu et al., 2021).

daily-average wind speeds as shown in Fig. 7. Another interesting analysis was done by (Pu et al., 2021) for inspecting brain functional connectivity of autism from blood-oxygenation-level-dependent time series as shown in Fig. 8. A recent work by (Imani & Shrivastava, 2023) utilized CI graphs for multivariate timeseries segmentation problem. They successfully demonstrated their technique on a physical activity monitoring dataset.

Running Graph Neural Networks over CI Graphs. Various GNN based techniques can be learnt over the CI graph. Especially, methods that are designed to run on Probabilistic Graphical models like the Cooperative Neural Networks (Shrivastava, Bart, Price, Dai, Dai, & Aluru, 2018), Bayesian Deep Learning methods (Wilson, 2020) and GNNs developed for other applications (Duvenaud, Maclaurin, Iparraguirre, Bombarell, Hirzel, Aspuru-Guzik, & Adams, 2015; Henaff, Bruna, & LeCun, 2015; Battaglia, Hamrick, Bapst, Sanchez-Gonzalez, Zambaldi, Malinowski, Tacchetti, Raposo, Santoro, Faulkner, et al., 2018) can be potentially adapted to work with CI graphs. A study of attribute propagation over CI graphs was presented in (Chajewska & Shrivastava, 2023). Neural Graphical Models (Shrivastava & Chajewska, 2023b) can potentially learn richer distributions compared to feature dependencies discovered in a Conditional Independence graph.

4. Conclusion

In this survey on the recently developed graph recovery methods, we attempted to build a case for Conditional Independence graphs for analysis of data from various domains. We provided a breakdown of different methods, traditionally used, as well as recently developed deep learning models, for CI graph recovery along with a primer on their implementation and functioning. In order to facilitate wider adoption, this work also provided various approaches and best practices to handle input data with mixed datatypes which is a critical preprocessing step, often tricky to manage. We laid out several use cases for CI graphs with the hope that they will become one of the mainstream methods for data exploration and insight extraction.

References

Aluru, M., Shrivastava, H., Chockalingam, S. P., Shivakumar, S., & Aluru, S. (2022). Engrain: a supervised ensemble learning method for recovery of large-scale gene regulatory networks. *Bioinformatics*, *38*(5), 1312–1319.

Baba, K., Shibata, R., & Sibuya, M. (2004). Partial correlation and conditional correlation as measures of conditional independence. *Australian & New Zealand Journal of Statistics*, *46*(4), 657–664.

Banerjee, O., Ghaoui, L. E., & d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, *9*(Mar), 485–516.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. (2018). Relational inductive biases, deep learning, and graph networks..

Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, *2*(1), 183–202.

Belilovsky, E., Kastner, K., Varoquaux, G., & Blaschko, M. B. (2017). Learning to discover sparse graphical models. In *International Conference on Machine Learning*, pp. 440–448. PMLR.

Bhattacharya, S., Rajan, V., & Shrivastava, H. (2017). ICU mortality prediction: a classification algorithm for imbalanced datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

Bhattacharya, S., Rajan, V., & Shrivastava, H. (2019). Methods and systems for predicting mortality of a patient.. US Patent 10,463,312.

Bollhofer, M., Eftekhari, A., Scheidegger, S., & Schenk, O. (2019). Large-scale sparse inverse covariance matrix estimation. *SIAM Journal on Scientific Computing*, *41*(1), A380–A401.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, *3*(1), 1–122.

Chajewska, U., & Shrivastava, H. (2023). Knowledge propagation over conditional independence graphs..

Chatrabgoun, H., Soltanian, A.-R., Mahjub, H., & Bahreini, F. (2021). Learning gene regulatory networks using gaussian process emulator and graphical lasso. *Journal of Bioinformatics and Computational Biology*, *19*(03), 2150007.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chen, T., Chen, X., Chen, W., Wang, Z., Heaton, H., Liu, J., & Yin, W. (2022). Learning to optimize: A primer and a benchmark. *The Journal of Machine Learning Research*, *23*(1), 8562–8620.

Chen, X., Liu, J., Wang, Z., & Yin, W. (2018). Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. In *Advances in Neural Information Processing Systems*, pp. 9061–9071.

Chen, Z., Tan, S., Chajewska, U., Rudin, C., & Caruana, R. (2023). Missing values and imputation in healthcare data: Can interpretable machine learning help?. In *Conference on Health, Inference and Learning (CHIL)*.

Danaher, P., Wang, P., & Witten, D. M. (2014). The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, *76*(2), 373.

Denton, E., & Fergus, R. (2018). Stochastic video generation with a learned prior. In *International Conference on Machine Learning*, pp. 1174–1183. PMLR.

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., & Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems*, pp. 2224–2232.

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, *9*(3), 432–441.

Greenewald, K., Zhou, S., & Hero III, A. (2019). Tensor graphical lasso (TeraLasso). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *81*(5), 901–931.

Gregor, K., & LeCun, Y. (2010). Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 399–406. Omnipress.

Guo, S., Jiang, Q., Chen, L., & Guo, D. (2016). Gene regulatory network inference using PLS-based methods. *BMC Bioinformatics*, *17*(1), 1–10.

Hallac, D., Park, Y., Boyd, S., & Leskovec, J. (2017). Network inference via the time-varying graphical lasso. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 205–213.

Haury, A.-C., Mordelet, F., Vera-Licona, P., & Vert, J.-P. (2012). TIGRESS: trustful inference of gene regulation using stability selection. *BMC Systems Biology*, *6*(1).

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, *20*(3), 197–243.

Heckerman, D., Horvitz, E., & Nathwani, B. N. (1992). Toward normative expert systems part I. The Pathfinder project. *Methods of Information in Medicine*, *31*, 90–105.

Heckerman, D., & Nathwani, B. N. (1992). Toward normative expert systems part II. Probability-based representations for efficient knowledge acquisition and inference. *Methods of Information in Medicine*, *31*, 106–116.

Heinze-Deml, C., Maathuis, M. H., & Meinshausen, N. (2018). Causal structure learning. *Annual Review of Statistics and Its Application*, *5*, 371–391.

Henaff, M., Bruna, J., & LeCun, Y. (2015). Deep convolutional networks on graph-structured data..

Hsieh, C.-J., Sustik, M. A., Dhillon, I. S., Ravikumar, P., et al. (2014). QUIC: quadratic approximation for sparse inverse covariance estimation.. *J. Mach. Learn. Res.*, *15*(1), 2911–2947.

Hsieh, C.-J., Sustik, M. A., Dhillon, I. S., Ravikumar, P. K., & Poldrack, R. (2013). BIG & QUIC: Sparse inverse covariance estimation for a million variables. *Advances in Neural Information Processing Systems*, *26*.

Imani, S., & Shrivastava, H. (2023). Are uGLAD? Time will tell!..

Jones, D. T., Buchan, D. W., Cozzetto, D., & Pontil, M. (2012). PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, *28*(2), 184–190.

Jung, A., Hannak, G., & Goertz, N. (2015). Graphical lasso based model selection for time series. *IEEE Signal Processing Letters*, *22*(10), 1781–1785.

Kaggle (2022). Lung Cancer. `https://www.kaggle.com/datasets/nancyalaswad90/lung-cancer?select=survey+lung+cancer.csv`.

Kalofolias, V. (2016). How to learn a graph from smooth signals. In *Artificial Intelligence and Statistics*, pp. 920–929. PMLR.

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Komodakis, N., & Pesquet, J.-C. (2015). Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems. *IEEE Signal Processing Magazine*, *32*(6), 31–54.

Liu, J., & Chen, X. (2019). ALISTA: Analytic weights are as good as learned weights in LISTA. In *International Conference on Learning Representations (ICLR)*.

Liu, Y., Palmedo, P., Ye, Q., Berger, B., & Peng, J. (2018). Enhancing evolutionary couplings with deep convolutional neural networks. *Cell Systems*, *6*(1), 65–74.

Loh, P.-L., & Wainwright, M. J. (2011). High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. *Advances in Neural Information Processing Systems*, *24*.

McLachlan, S., Dube, K., Hitman, G. A., Fenton, N. E., & Kyrimi, E. (2020). Bayesian networks in healthcare: Distribution by medical condition. *Artificial Intelligence in Medicine*, *107*, 101912.

Moerman, T., Aibar Santos, S., Bravo Gonzalez-Blas, C., Simm, J., Moreau, Y., Aerts, J., & Aerts, S. (2019). GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks. *Bioinformatics*, *35*(12), 2159–2161.

Niitsuma, H., & Lee, M. (2016). Word2vec is a special case of kernel correspondence analysis and kernels for natural language processing..

Pamfil, R., Sriwattanaworachai, N., Desai, S., Pilgerstorfer, P., Georgatzis, K., Beaumont, P., & Aragam, B. (2020). DYNOTEARS: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*, pp. 1595–1605. PMLR.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pu, X., Cao, T., Zhang, X., Dong, X., & Chen, S. (2021). Learning to learn graph topologies. *Advances in Neural Information Processing Systems*, *34*.

Rahman, M. M., & Davis, D. N. (2013). Addressing the class imbalance problem in medical datasets. *International Journal of Machine Learning and Computing*, *3*(2), 224.

Ravikumar, P., Wainwright, M. J., Raskutti, G., & Yu, B. (2011). High-dimensional covariance estimation by minimizing l_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, *5*, 935–980.

Rolfs, B., Rajaratnam, B., Guillot, D., Wong, I., & Maleki, A. (2012). Iterative thresholding algorithm for sparse inverse covariance estimation. *Advances in Neural Information Processing Systems*, *25*, 1574–1582.

Rothman, A. J., Bickel, P. J., Levina, E., Zhu, J., et al. (2008). Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, *2*, 494–515.

Sheskin, D. J. (2003). *Handbook of parametric and nonparametric statistical procedures*. Chapman and hall/CRC.

Shrivastava, G., & Shrivastava, A. (2020). Diverse video generation using a gaussian process trigger. In *International Conference on Learning Representations*.

Shrivastava, G., Shrivastava, H., & Shrivastava, A. (2020). Learning what not to model: Gaussian process regression with negative constraints..

Shrivastava, H. (2020). *On Using Inductive Biases for Designing Deep Learning Architectures*. Ph.D. thesis, Georgia Institute of Technology.

Shrivastava, H. (2023). Reconstruction of gene regulatory networks using sparse graph recovery models..

Shrivastava, H., Bart, E., Price, B., Dai, H., Dai, B., & Aluru, S. (2018). Cooperative neural networks (conn): Exploiting prior independence structure for improved classification. *Advances in Neural Information Processing Systems*, *31*.

Shrivastava, H., & Chajewska, U. (2023a). Neural Graph Revealers. In *Workshop on Machine Learning for Multimodal Healthcare Data (ML4MHD 2023) at Fortieth International Conference on Machine Learning (ICML 2023)*.

Shrivastava, H., & Chajewska, U. (2023b). Neural Graphical Models. In *Proceedings of the 17th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU), to appear*.

Shrivastava, H., Chajewska, U., Abraham, R., & Chen, X. (2022a). A deep learning approach to recover conditional independence graphs. In *NeurIPS 2022 Workshop: New Frontiers in Graph Learning*.

Shrivastava, H., Chajewska, U., Abraham, R., & Chen, X. (2022b). `uGLAD`: Sparse graph recovery by optimizing deep unrolled networks..

Shrivastava, H., Chen, X., Chen, B., Lan, G., Aluru, S., Liu, H., & Song, L. (2020). `GLAD`: Learning sparse graph recovery. In *International Conference on Learning Representations*.

Shrivastava, H., Huddar, V., Bhattacharya, S., & Rajan, V. (2015). Classification with imbalance: A similarity-based method for predicting respiratory failure. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 707–714. IEEE.

Shrivastava, H., Huddar, V., Bhattacharya, S., & Rajan, V. (2021). System and method for predicting health condition of a patient.. US Patent 11,087,879.

Shrivastava, H., Zhang, X., Aluru, S., & Song, L. (2020). GRNUlar: Gene regulatory network reconstruction using unrolled algorithm from single cell RNA-sequencing data..

Shrivastava, H., Zhang, X., Song, L., & Aluru, S. (2022). GRNUlar: A deep learning framework for recovering single-cell gene regulatory networks. *Journal of Computational Biology*, *29*(1), 27–44.

Sojoudi, S. (2016). Equivalence of graphical lasso and thresholding for sparse graphs. *The Journal of Machine Learning Research*, *17*(1), 3943–3963.

Spirtes, P., & Meek, C. (1995). Learning Bayesian networks with discrete variables from data.. In *KDD*, Vol. 1, pp. 294–299.

Stadler, N., & Buhlmann, P. (2012). Missing values: sparse inverse covariance estimation and an extension to sparse regression. *Statistics and Computing*, *22*(1), 219–235.

Sun, J., Li, H., Xu, Z., et al. (2016). Deep ADMM-Net for compressive sensing MRI. In *Advances in Neural Information Processing Systems*, pp. 10–18.

Sun, Q., Tan, K. M., Liu, H., & Zhang, T. (2018). Graphical nonconvex optimization via an adaptive convex relaxation. In Dy, J., & Krause, A. (Eds.), *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 of *Proceedings of Machine Learning Research*, pp. 4810–4817. PMLR.

United States Department of Health and Human Services, Division of Vital Statistics (DVS) (2015). *Birth Cohort Linked Birth – Infant Death Data Files, 2004-2015*. Compiled from data provided by the 57 vital statistics jurisdictions through the Vital Statistics Cooperative Program, on CDC WONDER On-line Database. Accessed at `https://www.cdc.gov/nchs/data_access/vitalstatsonline.htm`.

Van Anh Huynh-Thu, A. I., Wehenkel, L., & Geurts, P. (2010). Inferring regulatory networks from expression data using tree-based methods. *PloS one*, *5*(9).

Wang, B., Gao, J., & Qi, Y. (2017). A fast and scalable joint estimator for learning multiple related sparse Gaussian graphical models. In *Artificial Intelligence and Statistics*, pp. 1168–1177. PMLR.

Wang, Y., Jang, B., & Hero, A. (2020). The Sylvester graphical lasso (SyGlasso). In *International Conference on Artificial Intelligence and Statistics*, pp. 1943–1953. PMLR.

Williams, D. R. (2020). Beyond lasso: A survey of nonconvex regularization in Gaussian graphical models..

Wilson, A. G. (2020). The case for Bayesian deep learning..

Yang, E., Lozano, A. C., & Ravikumar, P. K. (2014). Elementary estimators for graphical models. *Advances in Neural Information Processing Systems*, *27*.

Yu, Y., Chen, J., Gao, T., & Yu, M. (2019). DAG-GNN: DAG structure learning with graph neural networks. In *International Conference on Machine Learning*, pp. 7154–7163. PMLR.

Zhang, M., Jiang, S., Cui, Z., Garnett, R., & Chen, Y. (2019). D-VAE: A variational autoencoder for directed acyclic graphs. *Advances in Neural Information Processing Systems*, *32*.

Zhang, R., Fattahi, S., & Sojoudi, S. (2018). Large-scale sparse inverse covariance estimation via thresholding and max-det matrix completion. In *International Conference on Machine Learning*, pp. 5766–5775. PMLR.

Zheng, X., Aragam, B., Ravikumar, P. K., & Xing, E. P. (2018). DAGs with NO TEARS: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, *31*, 9472–9483.

Zheng, X., Dan, C., Aragam, B., Ravikumar, P., & Xing, E. (2020). Learning sparse nonparametric DAGs. In *International Conference on Artificial Intelligence and Statistics*, pp. 3414–3425. PMLR.