Missing Values in Multiple Joint Inference of Gaussian Graphical Models

Veronica Tozzo[*] VTOZZO@MGH.HARVARD.EDU

Center for System Biology and Department of Pathology, Massachusetts General Hospital
Department of Systems Biology, Harvard Medical School

Davide Garbarino DAVIDE.GARBARINO@EDU.UNIGE.IT

Annalisa Barla ANNALISA.BARLA@UNIGE.IT

Department of Informatics, Bioengineering, Robotics and System Engineerings, Universitá di Genova

Abstract

Real-world phenomena are often not fully measured or completely observable, raising the so-called *missing data* problem. As a consequence, the need of developing ad-hoc techniques that cope with such issue arises in many inference contexts. In this paper, we focus on the inference of Gaussian Graphical Models (GGMs) from multiple input datasets having complex relationships (*e.g.* multi-class or temporal). We propose a method that generalises state-of-the-art approaches to the inference of both multi-class and temporal GGMs while naturally dealing with two types of missing data: partial and latent. Synthetic experiments show that our performance is better than state-of-the-art. In particular, we compared results with single network inference methods that suitably deal with missing data, and multiple joint network inference methods coupled with standard pre-processing techniques (*e.g.* imputing). When dealing with fully observed datasets our method analytically reduces to state-of-the-art approaches providing a good alternative as our implementation reaches convergence in shorter or comparable time. Finally, we show that properly addressing the missing data problem in a multi-class real-world example, allows us to discover interesting varying patterns.

Keywords: Missing data; Multiple joint network inference; Multi-class; Time-series

1. Introduction

Consider the two following scenarios: (i) during a medical trial we submit a survey to patients to assess their status, for privacy concerns they refuse to answer to some questions; (ii) during a weather monitoring experiment, we consistently do not measure the humidity in the air. These two types of situations lead to two different concepts: in the first case, questionnaires present missing values randomly positioned depending on the concerns of the patient, we call the final data *partial*. Differently, in the second scenario, we do not have any observation of humidity, thus we say that the related data are *latent* (Little and Rubin, 2019). We will generically refer to these types of data as *missing data*, to indicate one of the two or a combination of both.

Missing data require careful consideration in all machine learning and statistical settings. In this paper, we restrict to the problem of inferring multiple joint Gaussian Graphical Models (GGMs) when the input data may contain missing values. A multiple joint GGM is represented by a set of undirected graph $G = (G_k)_{k=1}^K = (V, E_k)_{k=1}^K$, where $V = \{X_1, \dots, X_d\}$ is a finite set of nodes that represent random variables, and each $E_k \subseteq V \times V$ is a set of edges, where E_k is not necessarily

[*] Work partially done while at Università di Genova.

equal to E_j for $k \neq j$. Each $k = 1, \dots, K$ defines a specific connectivity pattern and its meaning depends on the context. Either it could be associated with K different classes when dealing with multi-class network inference (Danaher et al., 2014; Guo et al., 2011) , or it could index a sorted sequence of discrete time-points when facing a problem of temporal network inference (Monti et al., 2014; Hallac et al., 2017; Tomasi et al., 2018). Each E_k univocally determines a multivariate normal distribution $\mathcal{N}(\mu_k, \Sigma_k)$. Indeed, the *precision matrix*, $\Theta_k = \Sigma_k^{-1}$ encodes the conditional independence between pairs of variables, *i.e.*, the structure of the graph G_k, since $\Theta_k(i, j)$ is 0 if and only if $(x_i, x_j) \notin E_k$ (Lauritzen, 1996) Therefore, one can interpret the precision matrix as the weighted adjacency matrix of G_k.

Typically the graph structure G_k is not known as we can only observe the behaviour of the single variables in the system. We indicate the n_k observations of the d variables as $X_k = (x_{i1}, \dots, x_{id})_{i=1}^{n_k} \in \mathbb{R}^{n_k \times d}$ which is a dataset that may contain missing values. Joint multiple network inference aims at learning a series of precision matrices $\mathbf{\Theta} = (\Theta_k)_{1 \leq k \leq K}$ that are both sparse and consistent with each other from these observations. The sparsity assumption is due to the combinatorial nature of the problem that requires to be constraint to have identifiability guarantees (Friedman et al., 2008). The consistency assumption fulfills the need of a similar structure among classes/time points that belong to the same underlying system (Guo et al., 2011; Danaher et al., 2014; Monti et al., 2014; Hallac et al., 2017; Foti et al., 2016; Tomasi et al., 2018).

In literature, the problem of inferring graphs from observations that contain missing data has been tackled in the single graph inference case. A non-convex method based on Expectation Maximization (EM)(Dempster et al., 1977) has been used to cope with the problem of partial data (Städler and Bühlmann, 2012; Little and Rubin, 2019) and latent data (Yuan, 2012). Differently, in (Chandrasekaran et al., 2010, 2011; Choi et al., 2011) they considered a convex sparse plus low-rank decomposition method to marginalise out the effect of latent data. The inference of multiple joint GGMs in presence of latent data was tackled in (Foti et al., 2016; Tomasi et al., 2018; Chang et al., 2019) using the sparse plus low-rank approach only in the case of temporal networks. To the best of our knowledge, a unified way of dealing with all types of missing data and types of joint inference does not exist.

Here, we provide a method that is able to deal with both partial and latent data in the case of multiple GGMs. The novelty of the approach consists in providing a unique way of handling different types of multiple joint inference methods as well as two types of missing data. Moreover, in the case of latent data, the method can provide more insights on the actual latent connections compared to previously published work. The method leverages on the EM algorithm to deal with missing data (Städler and Bühlmann, 2012; Yuan, 2012) and kernels to deal with different types of joint inference problems. Indeed, in Tomasi et al. (2020) they proposed a kernel method that, by suitable choices of the kernel, allows to consider simultaneously different types of complex temporal relationships and multi-class data.

Synthetic experiments show that our method provides good results in terms of estimate of the true underlying network as well as good estimate of the true values of the edges. We thoroughly evaluated the proposed method under several levels of complexity both in the multi-class and the temporal case. We also show that our method is able to retrieve possibly better results than the state-of-the-art in case of complete data. We also provide a real-world case study in which we compare the performances of the newly proposed method in a multi-class setting. We show that, differently from the state-of-the-art method coupled with imputing techniques we are indeed able to capture variability patterns within the classes that are consistent with the data.

Outline The rest of this paper is organised as follows. Section 2 introduces the reader to the definition of partial and latent variables and the related induced problems. Section 4 presents the method that allows for the inference of multiple GGMs from missing data. Section 5 presents the synthetic experiments on the proposed methods. Section 5.2 presents a real-world case study. Finally, Section 6 concludes with a discussion and future research directions.

2. Missing data

Missing data are values of a dataset that are un-observed and may possibly be meaningful for a specific analysis task. In this paper, we consider two possible patterns of data missing at random, *i.e.* the missing values does not depend on the value itself. Each of these two types of data introduces different problems during the inference of networks (Little and Rubin, 2019). Given k data matrices $X_k \in \mathbb{R}^{n_k \times d}$ which samples are assumed to be drawn from a multivariate Gaussian distribution, we formally define the two types of missing data as follows.

Partial data can be described as the absence of measurements randomly positioned in each observation. More formally, partial data consist in a set of matrices X_k where for each $k = 1, \dots, K$ and for each $i = 1, \dots, n_k$ there are some un-observed values indexed by the set $M_i^k \subseteq \{1, \dots, d\}$. These "holes" in the matrix make impossible direct computation and thus require pre-processing or ad-hoc inference mechanisms. The pre-processing approaches are typically: *complete cases* and *imputing*. In the complete cases we discard the samples that do not have complete measurements on the variables, while *imputing* filles the holes with a suitable value, *e.g.* the empirical mean. The issues deriving from complete cases and imputing pre-processing are the following: the former reduces the sample size drastically which may impede the correct inference of the underlying graph especially when $n \ll d$; the latter introduces substantial bias in the estimated solution even if it induces to believe that we can reason in terms of complete data (Little and Rubin, 2019; Madow et al., 1983). Hence, pre-processing techniques distort the empirical distribution of the variables which consequently leads to biased estimates of the underlying graph.

Latent data describe the consistent absence of some variable measurements across all samples. More formally, we observe d variables but there are l more, indexed by the set $M_i^k = \{d+1, \dots, d+l\}$ that are always unobserved across all data matrices $k = 1, \dots, K$ and samples $i = 1, \dots, n_k$. Therefore, on these variables, we do not have any information, not their number nor the relationship they have with the observed variables. Their presence in the system though, if not taken into account, leads to spurious edges, *i.e.*, links that would be conditioned away if the latent variables could have been observed (Chandrasekaran et al., 2010).

3. Joint multiple network inference

We consider a multiple joint Gaussian Graphical Model defined as a set of multivariate normal distributions $\mathcal{N}_G$ that factorises according to a set of graphs $G = (V, E_k)_{k=1}^K$. The inference of such graphs corresponds to the learning of the precision matrices $\boldsymbol{\Theta} = (\Theta_1, \dots, \Theta_k)$ that completely define the underlying distributions assuming that the means μ_k is zero for every k. The index k can have different meanings depending on the context. In this paper, we deal with the inference from: **multi-class data**, or *Joint Graphical Lasso* (JGL) (Guo et al., 2011; Danaher et al., 2014) where k indexes different classes of observations; **temporal data**, or *Time-varying Graphical Lasso* (TGL) (Hallac et al., 2017; Tomasi et al., 2018). where k corresponds to discrete time points obtained by dividing a time-series of length T in K chunks of equal size. In each chunk the samples are assumed to be *i.i.d.* The concept of joint inference lies in the fact that we guide the inference method with the prior that there is structural consistency among the k precision matrices $\mathbf{\Theta} = (\Theta_1, \dots, \Theta_k)$. In order to deal with both JGL and TGL with a unique inference method we recur to the methodology proposed in (Tomasi et al., 2020) where the authors exploit a kernel formulation to model the dependencies allowing to consider in a single way different joint inference problems. A kernel $\kappa \in \mathcal{S}^T_+$ is a positive semi-definite matrix that encodes, at each entry $\kappa(k, k')$, the strength of how much two different networks, Θ_k and $\Theta_{k'}$, should be similar in such a way that, samples belonging to different (but related) input matrices, can drive the inference toward a more reliable estimation of the structure especially when the number of samples is low.

Consider two networks indexed by k and k', the kernel models similarities dependency strength in such a way that a strength equal to zero ($\kappa[k, k'] = 0$) implies that graphs G_k and $G_{k'}$ are independent from each other and, therefore, no consistency is forced on them during the inference. In particular we will use for

- **multi-class data**, a kernel which enforces the same similarity on all classes

$$\kappa(k, k') = \begin{cases} 1, & \text{if } k = k' \\ \beta, & \text{if } k \neq k' \end{cases} \tag{1}$$

- **temporal data**, a kernel which enforces similarity only on consecutive time points

$$\kappa(t_i, t_j) = \begin{cases} 1, & \text{if } t_i = t_j \\ \beta, & \text{if } t_i = t_{j+1} \text{ or } t_i = t_{j-1} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Here, $\beta > 0$ is a constant value that measures the strength of how similar the graphs k and k' are.

In order to impose structure similarity that derives from joint inference, we adopt a penalised Maximum Likelihood Estimation (MLE) method where the penalisation is given by (Tomasi et al., 2020)

$$P_{\Psi,\kappa}(\mathbf{\Theta}) = \sum_{\substack{s,k=1\\ s>k}}^{K} \kappa_{sk}\Psi(\Theta_s - \Theta_k) = \sum_{k=1}^{K-1}\sum_{k'=1}^{K-k} \kappa_{kk'}\Psi(\Theta_{k+k'} - \Theta_{k'}). \tag{3}$$

Such penalty depends on the kernel κ that encodes the type of multiple joint network inference and the consistency function Ψ that defines the type of similarity to enforce on graphs that are dependent to each other according to the specified kernel.

In Hallac et al. (2017) the authors proposed different consistency functions. In this paper we make explicit use of the following two, but all the ones proposed in the original paper can be easily substituted. In particular, we will consider $\Psi(\cdot) = \ell_1(\cdot) = \sum_{ij} |\cdot|$, which is the lasso penalty that encourages few edges to change between subsequent time points while the rest of the structure remains the same Danaher et al. (2014) and $\Psi(\cdot) = \ell_2^2(\cdot) = \sum_{ij} \cdot_j^2$, which is the Laplacian penalty that causes smooth transitions (Hallac et al., 2015; Gibberd and Roy, 2017).

Note that substituting the explicit formulation of the kernels in Equations (1) and (2) in the penalty in Equation (3), leads to the original inference problems JGL (Danaher et al., 2014) and TGL (Hallac et al., 2017), respectively.

4. Joint network inference with missing data

For each k, consider a set of n_k observations $X_k \in \mathbb{R}^{n_k \times d}$ where each sample is a d-dimensional vector drawn from a multivariate Gaussian distribution $\mathcal{N}(\mu_k, \Theta_k)$ that may be not complete, *i.e.*, some values may be missing. We want to define a general network inference method that, from such observations, learns a set of precision matrices $\mathbf{\Theta} = (\Theta_1, \ldots, \Theta_K) \in \mathbb{R}^{(d \times d) \times K}$ and means $\mu = (\mu_1, \ldots, \mu_K) \in \mathbb{R}^{d \times K}$. Differently from literature, we need to estimate the means instead of assuming data to be centered in zero. Indeed, in presence of partial data we cannot compute the empirical mean and re-centre the data without introducing bias as we would distort the empirical distribution in the same way as imputation does (Little and Rubin, 2019).

The modelling and the inference of GGMs from these type of data is aid by the factorisation properties that hold for the multivariate normal distribution. Consider, for each k and $i = 1, \ldots, n_k$ the sets O_i^k and M_i^k of indices of observed and missing variables, respectively. Such sets allows us to divide the sample i as $X_k[i, :] = \big(X_k[i, O_i^k], X_k[i, M_i^k]\big)$. Accordingly, for each sample i it is possible to define block precision matrices that group the set of observed and missing variables together. This grouping procedure allows to obtain a conditional distribution that is still a multivariate normal distribution with parameters analytically related to the original ones (Little and Rubin, 2019). Thus, the resulting precision matrix Θ_k, for $k = 1, \ldots, K$ is

$$\Theta_k = \Sigma_k^{-1} = \left[\begin{array}{c|c} \Theta_k[M_i^k] & \Theta_k[M_i^k O_i^k] \\ \hline \Theta_k[O_i^k M_i^k] & \Theta_k[O_i^k] \end{array}\right], \tag{4}$$

and mean vectors are $\mu_k = (\mu_k[M_i^k], \mu_k[O_i^k])$.

The inference problem can then be defined as MLE. Note that, the likelihood has the same form for each k, and it is defined exploiting information on the observed data only (Städler and Bühlmann, 2012; Yuan, 2012). Nonetheless, the inference aims at learning all parts of the precision matrices Θ_k. The log-likelihood writes out as follows:

$$\ell(X_k[O^k] | \Theta_k) = \frac{1}{2} \sum_{i=1}^{n_k} \bigg(\log \det\big(\Theta^{-1}_{kO_i^k}\big) + \big(X_{kO_i^k} - \mu^{\top}_{kO_i^k}\big)\big(\Theta^{-1}_{kO_i^k}\big)^{-1}\big(X_{kO_i^k} - \mu_{kO_i^k}\big)\bigg). \tag{5}$$

Finally, given the likelihood, the kernel κ and a behaviour Ψ the functional to optimise to perform inference takes the form

$$\underset{\mathbf{\Theta} \in \mathcal{S}_+^d}{\text{minimize}} \sum_{k=1}^{K} \Big[-\ \ell(X_k[O^k] | \Theta_k) + \alpha \|\Theta_k\|_{\text{od},1} \Big] + P_{\Psi,\kappa}(\mathbf{\Theta}) \tag{6}$$

where $\| \cdot \|_{od,1}$ is the off-diagonal ℓ_1-norm, which promotes sparsity in the precision matrices and whose strength is regulated by α and the constraint $\Theta_k \in \mathcal{S}_+^d$ restricts the search space to the cone of positive definite matrices.

4.1 Latent data specialization

The optimisation problem presented in Equation (6) requires further considerations when in presence of latent data. Indeed, when introducing latent variables while the model remains the same we naturally introduce a further hyper-parameter. Consider the set M of missing values, in the case

Algorithm 1 EM algorithm

```
Inputs: $\Psi$ consistency function, $\kappa$ temporal dependencies, $\boldsymbol{X}$ samples,
$\alpha$ sparsity hyper-parameters
for $\iota = 1, \dots,$ do
    for $k = 1, \dots K$ do
        $c_{ki} = \mu_k[M_i^k] + K_k[M_i^k]^{-1}\Theta_k[M_i^k O_i^k](X_k[:, O_i^k] - \mu_t[O_i])$
        $\mathbb{E}[X_k[iv]|X_k[O_i^k], \mu_t^{\iota-1}\Theta_k^{\iota-1}] = \begin{cases} X_k[iv] & \text{if } v \in O_i^k \\ c_k[iv] & \text{if } v \in M_i^k \end{cases}$
        $\mathbb{E}[X_k[iv]X_k[iv']|X_k[O_i^k], \mu_t^{\iota-1}\Theta_k^{\iota-1}] = \begin{cases} X_k[iv]X_k[iv'] & \text{if } v, v' \in O_i^k \\ X_k[iv]c_k[iv'] & \text{if } v \in O_i^k \\ (\Theta_k[M_i^k]^{-1})_{vv'} + c_k[iv]c_k[iv'] & \text{otherwise} \end{cases}$
        $C_k^\iota[vv'] = \sum_{i=1}^{n_k} \mathbb{E}[X_k[iv]X_k[iv']|X_k[O_i^k], \mu_k^{\iota-1}\Theta_k^{\iota-1}]$
    for $k = 1, \dots K$ do
        $\mu_k^\iota = \frac{1}{n_k}(\sum_{i=1}^{n_k} X_k[i1], \dots, \sum_{i=1}^{n_k} X_k[id])$
    $\boldsymbol{\Theta}^\iota = \underset{\boldsymbol{\Theta} \succ 0}{\operatorname{argmin}} \sum_{k=1}^{K}[-n_k \ell_{GGM}(\Theta_k|C_k^\iota) + \alpha\|\Theta_k\|] + P_{\Psi,\kappa}(\boldsymbol{\Theta})$
```

of latent variables such set is assumed to be the same across all k multiple graphs and across all n_k samples. Nonetheless, no prior information on the number of latent variables is naturally provided with the data. The number of latent variables is exactly the cardinality of the set$|M| = r$, and thus, by imposing a certain set M we are imposing a certain number of latent variables. Such hyper-parameter can be selected via model selection strategies, but it can be shown that results are not particularly subsceptible to this choice, i.e., we reach similar performances even if the value r is not precisely the same of the true underlying system (Yuan, 2012).

The proposed model allows us to infer the relationships among latent variables and between latent and observed variables, thus obtaining an explicit information of them differently from what was done in Tomasi et al. (2018).

4.2 Optimisation

Problem (6) is non-convex because of the unknown values of the input data matrices $\boldsymbol{X}$. In the ideal case of complete input data, the estimation of the two sets of parameters $\boldsymbol{\mu}$, $\boldsymbol{\Theta}$ would be straightforward using optimisation methods as the Alternating Direction Methods of Multiplier (ADMM) (Boyd et al., 2011). Nonetheless, the lack of knowledge on input data requires a specific optimization method. We recur to the Expectation Maximization (EM) algorithm, an alternating procedure that has guarantees of reaching a local minimum of functional (6). The procedure consists of two steps the E-step and the M-step, both repeated at each iteration. In the former we compute the expectation on the missing values (thus estimating the complete $\boldsymbol{X}$) while in the latter we substitute the estimated complete data in the functional and we maximise it to retrieve distribution parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Theta}$. The EM algorithm for the complete data case is described in Algorithm 1. Such algorithm may get stuck in local optima, for this reason we may require multiple initialisations in order to detect the final best reliable network.

The E-step can be performed separately for each k because of the linearity of the expectation operator. In order to easily compute the expectation we express the likelihood in Equation (5) in terms of sufficient statistics of the Normal distribution. For each k we have two sufficient statistics: the sample mean μ_k^C and the sample covariance matrix $C_k = \frac{1}{n_k} X_k^\top X_k$. The E-steps consists of the estimation of the expectation of the two sufficient statistics μ_k^C and C_k, this can be done by observing that, thanks to the factorization property of the normal distribution, for each sample i the missing values indexed by M_i^k are again distributed according to a multivariate normal distribution. Thus, the expectation of the missing values can be computed as the mean of such distribution. By substituting the previous estimates for Θ_k and μ_k at iteration t we can compute the expectation as in line 6 of Algorithm 1 Given the estimates of the E-step we can now reason in terms of complete data, and the maximisation problem corresponds to the model proposed in Tomasi et al. (2020) and has the form in line 13 of Algorithm 1.

We call the implementation of this EM algorithm *Missing Multiple Graphical Lasso* (MMGL$_\kappa$), we typically omit the subscript κ if it is clear from the context. In general, it can either model the temporal graphical lasso or the joint graphical lasso by substituting kernels in Equations (1) and (2). Note that in Tomasi et al. (2020) they proposed different kernels also for the modelling of complex temporal patterns. In this paper, we do not mention them as it is not the aim of our work but we can substitute also these kernels to obtain more complex temporal patterns (*e.g.* seasonality).

5. Experimental validation

We assessed the performance of the proposed method in different scenarios compared with the state-of-the-art methods. Due to space restrictions, we refer the reader to the publicly available open source Python library `regain`[1] for complete details on the experiments. We designed three synthetic experiments:

- **Exp-PT**: we considered a temporal evolving graphical model (kernel of Equation (2)) in presence of partial data. We compared the performance with the time-varying graphical lasso (TGL) (Hallac et al., 2017) coupled with pre-processing techniques (imputing and complete data strategies) and the missing graphical lasso (MGL) (Städler and Bühlmann, 2012) that copes with partial data, but does not consider time (*i.e.*, we fit a model for each time point).
- **Exp-PC**: we considered a multi-class graphical model (kernel in Equation (1)) and we assessed performances of the state-of-the art method the Joint Graphical Lasso (JGL) (Danaher et al., 2014) coupled with imputing and complete data.
- **Exp-LT** we consider a latent temporal evolving graphical model (kernel specified in Equation (2)) and we compared the performance with the state-of-the-art method for latent time-varying graphical inference (LTGL) (Tomasi et al., 2018) and a static version of an EM approach, LVGLASSO, that deals with latent variables (Yuan, 2012).

For all these experiments we performed the analysis at five increasing values for the of observed variables $|O|$ in the interval $[10, 100]$. For each of these values we performed the experiments at three different percentages of missing data $5, 10, 20\%$ for both partial and latent cases. We fixed the number of samples $N_k = 100$ in order to consider an increasing ill-posedness as the number of observed variables increases. We fixed the number of classes/times $K = 10$, so that the total number of unknowns is given by the formula $K * \frac{|O|*(|O|-1)}{2}$. For each of these possibilities we repeated the experiments 10 times to have mean results.
Given the great number of total experiments and hyper-parameters for each considered method

[1] `https://github.com/veronicatozzo/regain`

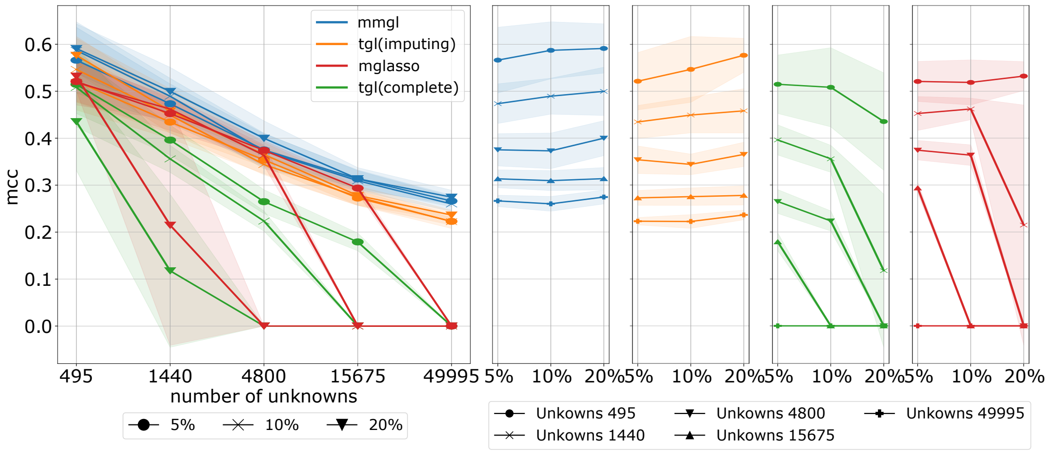


Figure 1: MCC of the results obtained for MMGL, TGL(complete), TGL(inputing) and MGL. Left panel all results, right panel mean results per percentage of partial data.

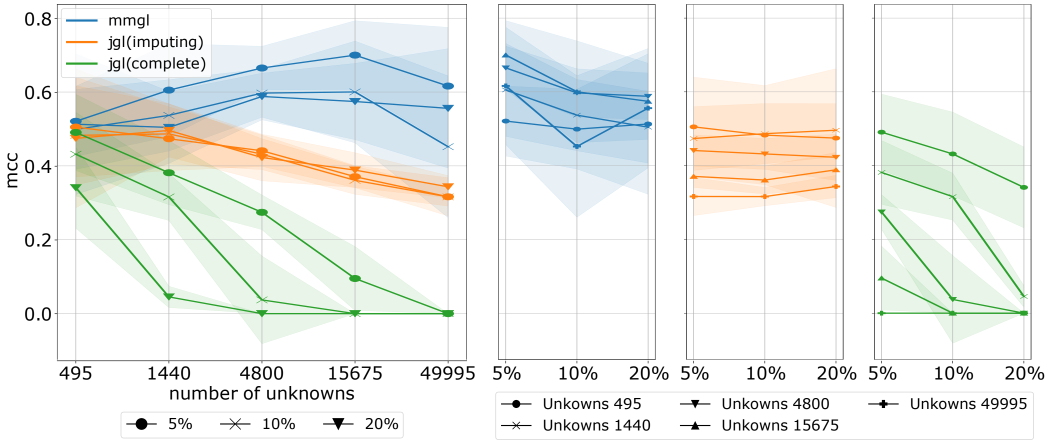


Figure 2: MCC of the results obtained for MMGL, JGL(complete), JGL(inputing). Left panel all results, right panel mean results per percentage of partial data.

performing ad-hoc cross-validation, even bayesian optimization method (Pelikan et al., 1999) would have required too much computational time (Tozzo and Barla, 2019). Thus, we fixed some of them to the value provided by theoretical bounds, while we fixed the others to reasonable arbitrary values. The hyper-parameters are as follows. All the models have the sparsity enforcing parameter α, MMGL has κ which, if we use either Equation (2) or (1), it coincides with the choice of a parameter β similarly to TGL (Hallac et al., 2017), JGL (Danaher et al., 2014) and LTGL (Tomasi et al., 2018). In the specific case of latent variables we also need to fix the number of latent variables r, similarly to LVGLASSO (Yuan, 2012). Lastly, LTGL has two further hyper-parameters τ and η that guide the latent variables behaviour in time. We fixed $\alpha = \frac{\log |O|}{N_k}$ (Raskutti et al., 2009) and $\tau = 2\sqrt{|O|/N_k}$ (Chandrasekaran et al., 2011) according to theoretical results, while we fixed $\beta = 1$ and r to the real number of latent variables in synthetic data. While one may object that fixing r is

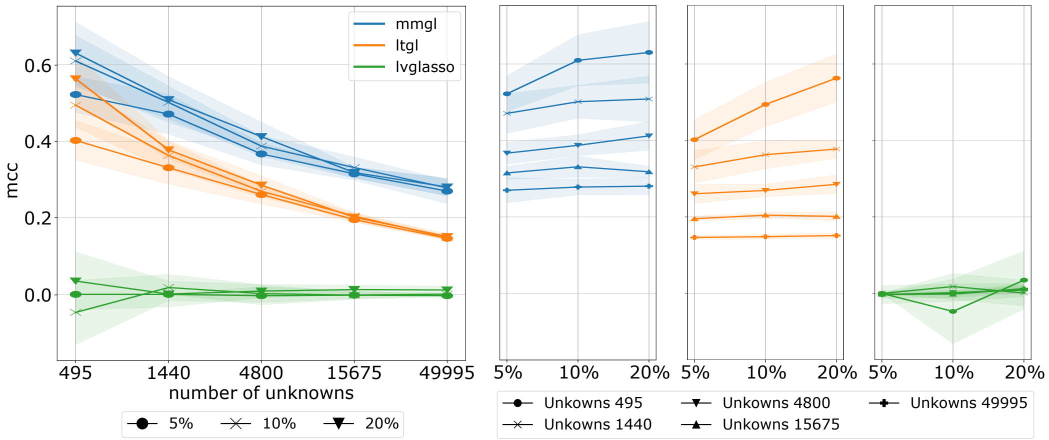


Figure 3: MCC of the results obtained for MMGL, LVGLASSO and LTGL. Left panel all results, right panel mean results per percentage of partial data.

not fair w.r.t. the choice of $\tau = 1$ for LTGL it can be shown that the choice of r does not impact heavily the performance of our algorithm and thus we opted to fix it to the true value. Such behaviour is similarly reported in (Yuan, 2012). Simulation results can be found in the github repository. All hyper-parameters were fixed to the same values for all the models under consideration.
We present the results in terms of Matthews Correlation Coefficient (MCC) (Matthews, 1975) where we considered as true positive the number of edges correctly identified, true negative the number of edges that were correctly inferred as absent, false positive is the number of edges that the algorithm identifies as existing but are not and false negative are those edges that are not inferred by the algorithm but exist.

5.1 Results

Results are presented in Figure 1, 2 and 3 for **Exp-PT**, **Exp-PC** and **Exp-LT** respectively. It can be seen that MMGL is the one with the highest MCC across all types of experiments while being stable as the percentage of partial or latent data increases (right panel of all three figures). The only state-of-the-art method that provides closely related performance is TGL with imputing as pre-processing strategy. When using complete data as pre-processing strategies, as well as when exploiting static method, the percentage of missing data matters on the final result. Indeed, regularization for multiple join inference allows us to exploit information on other classes/time to improve the inference as it can be seen also for TGL and JGL with imputing. In Figure 1 and 3, we observed that 20% missing data result in better performances, while not having a theoretical explanation for this behaviour we argue that it might simply be due to the data generation process. We performed also a scalability assessment in terms of convergence time. Results are presented in Figure 4, where we can observed that MMGL is the one that requires the highest time to converge in the **Exp-PT** and **Exp-LT**, this is expected as MMGL solves a non-convex problems that also estimates the means and precision matrices in time. Instead, in **Exp-PT** we can observe that it is faster than the available state-of-the-art implementation of JGL.

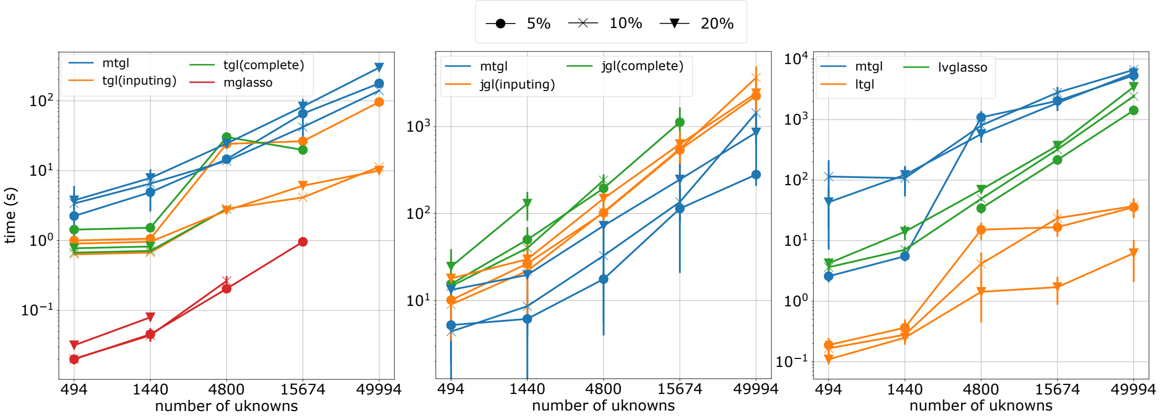


Figure 4: Scalability results for the experiments **Exp-PT** (left panel), **Exp-PC** (central panel) and **Exp-LT** (right panel).

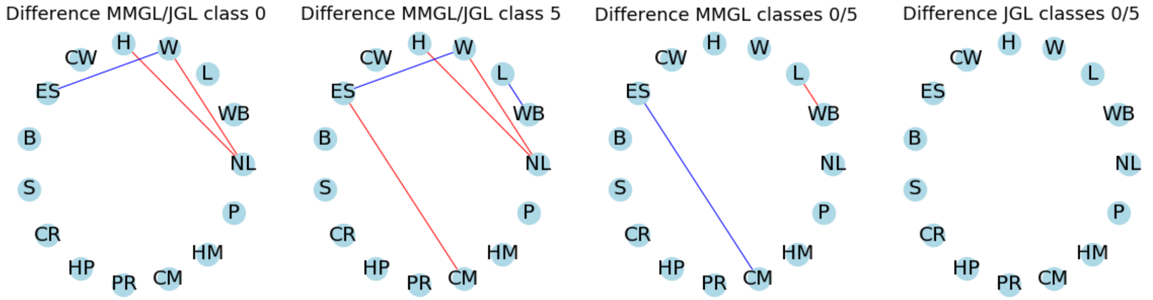
5.2 Case study: automobile safety

We performed a last experiment on a real-world dataset that consists of a multi-class dataset of automobile characteristics[2] where class 0 corresponds to set of safest cars and class 5 to the most risky sett. The dataset contains $\approx 26\%$ of missing values across all classes, which are also unbalances as we have 3 samples for class 0, 22 for class 1, 67 for class 2, 54 for class 3, 32 for class 4, and 27 for class 5. The hyper-parameters were selected through likelihood-based cross-validation in the following ranges $\alpha = [10^{-2}, 10^2]$ and $\beta = [10^{-5}, 10^1]$ and the resulting hyper-parameters are $\alpha = 10$ and $\beta = 10^{-4}$. We analysed the dataset with MMGL and JGL coupled with imputing, results are presented in Figure 5. The first thing that we can notice is that there are difference in the edges inferred in class 0 or 5 with the two methods as well as there are differences in the graph inferred with MMGL from the two classes but JGL tends to infer the same graph throughout all the classes. We do not discuss the possible insights obtained by this analysis as they are not the scope of this paper, but we want to remark that by exploiting MMGL we can obtain insights on the dataset that were impossible to obtain using JGL as it infers a network that is consistent across all five classes.

6. Conclusions

We presented a method that allows in a unique way to handle the inference of networks from data that may have different patterns of missing data as well as different complex inter-relationships as multi-class or temporal evolution. We showed on synthetic data that our method performs better than a variety of state-of-the-art methods thus providing a valuable alternative to both joint network inference method (Danaher et al., 2014; Hallac et al., 2017; Tomasi et al., 2018) as well as static inference methods that deal with missing data (Yuan, 2012; Städler and Bühlmann, 2012). MMGL optimization could be improved. Indeed, we could reduce time to converge by parallelising the K expectations computations. We do not provide such results but we expect the time to convergence to be reduced drastically. Moreover, in the case of latent variables we can reason in terms of blocks when computing the expectation step and we can remove the estimate of the means as re-centering

[2] `https://sci2s.ugr.es/keel/dataset.php?cod=90`


Difference MMGL/JGL class 0

Difference MMGL/JGL class 5

Difference MMGL classes 0/5

Difference JGL classes 0/5

Figure 5: Comparison of the networks obtained with MMGL and JGL on the automobile dataset. A blue edge means that the edge was present in the first named set and not in the second. Red edge the opposite. The labels correspond to Normalized-losses (NL), Wheel-base (WB), Length (L), Width (W), Height (H), Curb-weight (CW), Engine-size (ES), Bore (B), Stroke (S), Compression-ratio (CR), Horsepower (HP), Peak-rpm (PR), City-mpg (CM), Highway-mpg (HM), Price (P).

the data does not induce any bias. This approach is equivalent but would further reduce convergence time. Our derivations show that indeed handling latent data or partial data can be done in a unique way, and this show that the two different methods presented in (Yuan, 2012) for latent data and (Städler and Bühlmann, 2012) for partial data are, indeed, the same. As proposed in (Städler and Bühlmann, 2012), this method could be exploit as a pre-processing step to perform imputing and thus preparatory to supervised machine learning tasks. We leave the validation of this approach to further work due to lack of space. Also, our method different from methods that marginalise out the latent variables effect as (Chandrasekaran et al., 2010; Tomasi et al., 2018) and, by allowing a direct estimate of the latent variables graph may be used for complex inference and data mining.

References

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

V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky. Latent variable graphical model selection via convex optimization. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1610–1613. IEEE, 2010.

V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

A. Chang, T. Yao, and G. I. Allen. Graphical models and dynamic latent factors for modeling functional brain connectivity. In *2019 IEEE Data Science Workshop (DSW)*, pages 57–63. IEEE, 2019.

M. J. Choi, V. Y. Tan, A. Anandkumar, and A. S. Willsky. Learning latent tree graphical models. *Journal of Machine Learning Research*, 12(May):1771–1812, 2011.

P. Danaher, P. Wang, and D. M. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397, 2014.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

N. J. Foti, R. Nadkarni, A. Lee, and E. B. Fox. Sparse plus low-rank graphical models of time series for functional connectivity in meg. In *2nd KDD Workshop on Mining and Learning from Time Series*, 2016.

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

A. J. Gibberd and S. Roy. Multiple changepoint estimation in high-dimensional gaussian graphical models. *arXiv preprint arXiv:1712.05786*, 2017.

J. Guo, E. Levina, G. Michailidis, and J. Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15, 2011.

D. Hallac, J. Leskovec, and S. Boyd. Network lasso: Clustering and optimization in large graphs. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 387–396. ACM, 2015.

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

S. L. Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

R. J. Little and D. B. Rubin. *Statistical analysis with missing data*, volume 793. Wiley, 2019.

W. G. Madow, H. Nisselson, and I. Olkin. Incomplete data in sample surveys. vol. 1: Report and case studies. 1983.

B. W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

R. P. Monti, P. Hellyer, D. Sharp, R. Leech, C. Anagnostopoulos, and G. Montana. Estimating time-varying brain connectivity networks from functional mri time series. *NeuroImage*, 103:427–443, 2014.

M. Pelikan, D. E. Goldberg, E. Cantú-Paz, et al. Boa: The bayesian optimization algorithm. In *Proceedings of the genetic and evolutionary computation conference GECCO-99*, volume 1, pages 525–532. Citeseer, 1999.

G. Raskutti, B. Yu, M. J. Wainwright, and P. K. Ravikumar. Model selection in gaussian graphical models: High-dimensional consistency of ℓ_1-regularized mle. In *Advances in Neural Information Processing Systems*, pages 1329–1336, 2009.

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

F. Tomasi, V. Tozzo, S. Salzo, and A. Verri. Latent variable time-varying network inference. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2338–2346. ACM, 2018.

F. Tomasi, V. Tozzo, and A. Barla. Temporal pattern detection in time-varying graphical models. *Under review*, 2020.

V. Tozzo and A. Barla. Multi-parameters model selection for network inference. In *International Conference on Complex Networks and Their Applications*, pages 566–577. Springer, 2019.

M. Yuan. Discussion: Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40(4):1968–1972, 2012.