Midterm Update: Learning Gaussian Graphical Models with Temporal Dynamics

Graham Collins (ISYE), Jakob Krzyston (ECE), Sivabalan Manivasagam (CS), Matt O'Shaughnessy (ECE)

Project summary. One major section of our probabilistic graphical models course is understanding the algorithms used to learn the graphical representation from data. A popular graphical model representation used to understand real-world, high-dimensional data is the gaussian graphical model (GGM). The GGM provides insights into the conditional independences and relationships between observed variables through the inverse covariance matrix, also called the precision or concentration matrix. One method used to learn GGMs from observed data is the graphical lasso algorithm [5]. The graphical lasso is a convex optimization problem which optimizes fidelity of the sample precision matrix to the observed data while encouraging sparsity through ℓ_1 regularization by solving $\boldsymbol{K}_O^* = \underset{\boldsymbol{K}_O^* \succeq 0}{\arg\min}\ -\log\det \boldsymbol{K}_O^* + \operatorname{Tr}[\Sigma_O^* \boldsymbol{K}_O^*] + \lambda \|\boldsymbol{K}_O^*\|_1$, where $\boldsymbol{K}_O^*$ is the recovered precision matrix and Σ_O^* is the sample observed covariance matrix.

One drawback of the graphical lasso method is that, despite using a sparsity-encouraging objective function, the learned GGM is often extremely dense and uninterpretable. In [2], Chandrasekaran et al. overcome this issue by hypothesizing that the connections in the observed variables can be explained by a few unobserved, or latent, variables. They decompose the concentration matrix ($\boldsymbol{K}_O^*$) into a sparse observable term $\boldsymbol{S}$ and a low-rank hidden term $\boldsymbol{L}$ and the solve a similar convex optimization problem, encouraging the sparse and low-rank structure in $\boldsymbol{S}$ and $\boldsymbol{L}$ with ℓ_1 and nuclear-norm penalties, respectively.

Chandrasekaran et al. assumes the observed data samples are independent and identically distributed. Foti et al. extend this approach to stationary time series data [4]. The observed data is transformed into the frequency domain, where a spectral density matrix is learned using the Whittle likelihood approximation and group ℓ_1 lasso penalty across frequencies to find structure.

Although [4] effectively treats time series inputs, it assumes that the underlying GGM is not changing over time (stationarity assumption). After implementing existing methods, we plan to extend these algorithms to the case where we are interested in *tracking* a GGM using dynamic time series data.

State of the Field

GLASSO introduced by Friedman In order to learn sparse graphs, Friedman applies lasso penalty to the inverse covariance matrix. Graphical lasso uses coordinate descent to efficiently learn graphs, and it is now used widely (with additional developments). Lasso is a convex optimization problem so there are many ways to solve for global optima. Lasso makes models more interpretable and improves accuracy, and this is achieved by adding the constraint that the regression coefficients do not exceed some fixed value, urging coefficients to be zero. Ridge regression uses l2 norm to reduce coefficients, which slightly encourages sparsity. By using the l1 norm, we encourage as much sparsity as possible while still keeping the problem convex for faster/ absolute guarantees. Optimization problem: $\boldsymbol{K}_O^* = \underset{\boldsymbol{K}_O^* \succeq 0}{\arg\min}\ -\log\det \boldsymbol{K}_O^* + \operatorname{Tr}[\Sigma_O^* \boldsymbol{K}_O^*] + \lambda \|\boldsymbol{K}_O^*\|_1$, where $\boldsymbol{K}_O^*$ is the recovered precision matrix and Σ_O^* is the sample covariance matrix.The algorithm goes as follows:

- First, begin with $W = \Sigma_O^* + \rho I$ During the following steps, the diagonal of W is unchanged.
- For each $j = 1, .2, ...p, 1, 2, ...p, ...$, solve the lasso problem (below), which uses inner products W_{11} and s_{12}. This gives a $p-1$ vector $\hat{\beta}$. Fill in the corresponding row and column of W using $w_{12} = W_{11}\hat{\beta}$. Lasso problem is $\min_\beta\ \frac{1}{2}\|W_{11}^{1/2}\beta - b\|_2^2 + \rho\|\beta\|_1$
- Repeat until converges.

Sparse plus low rank graphical models of time series for functional connectivity in MEG For many applications, it would be useful to learn graphical models from high dimensional data. Examples include functional brain connectivity, stock, symptom networks, and more. Many previous methods ignored temporal dynamics, i.e. they assumed independent and identically distributed random variables through time. Additionally, most models only represent explicitly relevant variables, which means they ignore and miss important relationships. By including latent variables to represent random variables that are not explicitly defined, more accurate conditional independencies can be discovered. As such, the Fox paper addresses the challenge by learning Gaussian graphical models of time series with latent processes. They allow for heterogeneity between groups by using a hierarchal penalty.

Overall, they define the task as a convex optimization problem which is then solvable (using alternating directions method of multipliers algorithm in the paper). By encouraging sparsity, methods discover true relationships in an interpretable way. Such methods are able to eliminate edges and include relationships closer to the true structure of node relationships, defining a more conditionally independent structure of the random variables. The key application from the paper was brain connectivity from MEG, and the new method allowed for the discovery of more conditional independencies to learn about structure and connections in the brain. Inclusion of extra edges obscures the true relationships and makes graphs uninterpretable.

Inspired by likelihood-based approaches that transfer into the frequency domain with Bayesian priors on the inverse spectral density matrices, the Fox group accounts for unobserved variables with latent variables. When the latent variables are left out of the model, methods include spurious edges between nodes. Instead, the Fox paper uses ℓ_1 penalized graphical lasso to learn model structure, while including latent variables to reduce edges and have group structure to make use of independencies across frequencies. When we are not able to find conditional independencies, we cannot interpret the relationships of the graph. Disconnected nodes could appear connected if latent processes are not represented in the model.

Like glasso, Fox uses a penalty to force decomposition of inverse spectral density matrices into sparse/low rank parts with similarities across frequencies. They use ADMM to deal with the computational challenge of high number of frequencies in the problem. With these methods, they demonstrated learning of sparse graphs that make intuitive sense based on known connections (about the global economy for the stock example). They also learned neural connectivity relationships. While the method cannot find all independencies in the noisy data, the approach removes a lot of false edges without introducing many non-existent independencies (which would be very problematic).

Latent Variable Graphical Model Selection Via Convex Optimization In the work done by Chadrasekaran et al. [2], two main assumptions made were the variables in the graph are jointly Gaussian, and the resulting graphical model should be sparse. In this study, a tractable convex program based on regularized maximum-likelihood for model selection is used to learn latent variables. Using dimensionality reduction, in addition to graphical models, the regularizer uses an l_1 norm and nuclear norm to address the challenge of estimating the covariance matrices in high-dimensional settings.

The foundation of the work in is the context of Gaussian graphical models (GGM). The structure of the GGM is typically drawn from the structure of the concentration matrix $\boldsymbol{K}_{OH}^*$, which is the inverse of the covariance matrix $(\Sigma_{OH}^*)^{-1}$. The concentration matrix is dependent upon observed and hidden variables, O and H respectively. The covariance matrix is formed as a sum of a sparse matrix and a low rank matrix.

The marginalization of the concentration matrix is determined by the Schur Complement for GGM with latent variables is shown below:

$$\widetilde{\boldsymbol{K}}_O^* = (\Sigma_O^*)^{-1} = \boldsymbol{K}_O^* - \boldsymbol{K}_{O,H}^*(\boldsymbol{K}_H^*)^{-1}\boldsymbol{K}_{H,O}^*$$

Where $\widetilde{\boldsymbol{K}}_O^*$ is the marginal observed concentration matrix, $\boldsymbol{K}_O^*$ is the concentration matrix of observed variables conditioned on the latent variables, i.e., the top-left matrix block of $\boldsymbol{K}_{(O,H)}^*$. We assume $\boldsymbol{K}_O^*$ (equivalent notation to $\boldsymbol{S}$) is sparse. $\boldsymbol{K}_{O,H}^*(\boldsymbol{K}_H^*)^{-1}\boldsymbol{K}_{H,O}^*$ (equivalent notation to $\boldsymbol{L}$) is called a summary term of the effect of marginalization over hidden variables. This term is low rank when we assume view hidden variables relative to number of observed. This decomposition produces a graphical structure which is composed of a sparse term (graphical-component) and low-rank hidden term (PCA-like hidden component).

Dynamic Filtering of Time-Varying Sparse Signals via l1 Minimization To deal with high-dimensional streaming data, algorithms are needed for dynamic filtering of time-varying sparse signals. BPDN-DF basis pursuit denoising dynamic filtering lacks performance guarantees but has performed well when system dynamics are known. Charles et. al present a new algorithm, RWL1-DF, that is robust to model inaccuracies without adding significant computational complexity. The hierarchical probabilistic data model and carrying of statistics from prior estimates allows for this development (in a similar way to Kalman filtering)

In dynamic filtering, the system is estimated from the previous state (lag 1, for example). Kalman filtering, a popular approach, assumes normal distributions and linearity. Charles et al takes advantage both of the sparse nature of the model and of the system dynamics using l1 optimization as well as higher-order statistics from the system's previous time step. BPDN-DF is efficient and works well when system dynamics are known. However, it does not work when the system dynamics are not well behaved/defined. The new method (RWL1-DF) is able to perform well regardless thanks to a hierarchical probabilistic data model and utilization of statistics from the previous estimate.

Reweighted ℓ_1 dynamic filtering for tracking. In 2008, Candès et al. introduced the reweighted ℓ_1 (RWL1) algorithm for sparse estimation. Intuitively, RWL1 addresses a deficiency of ℓ_1 based methods (e.g., BPDN, LASSO): the ℓ_1 norm (used as a convex proxy for the ideal ℓ_0 penalty) penalizes nonzero coefficients with penalties proportional to their magnitudes. RWL1 addresses this deficiency by introducing weights on nonzero elements proportional to their estimated magnitudes. The RWL1 algorithm consists of alternating between updating the weights and estimated nonzero magnitudes; convergence typically occurs in just a few iterations.

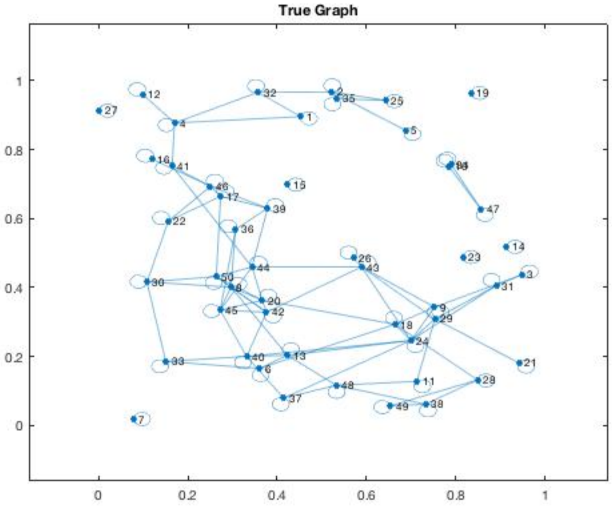
Adopting a probabilistic view of the LASSO and RWL1 problems allow us to extend the idea of reweighting from a method for improving sparse coding algorithms to a general framework for incorporating prior knowledge into sparse recovery. From a probabilistic perspective, the LASSO is a MAP estimator using a Gaussian likelihood and Laplacian prior — which is known to encourage sparsity because of its strong peak at zero and heavy tails. Intuitively, this property of the Laplacian prior encourages most elements in the recovered vector to be zero, while still allowing some elements to be nonzero when they are supported by enough evidence.

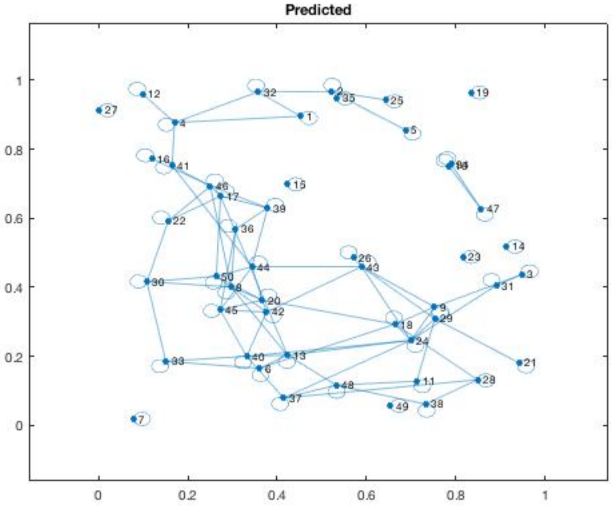
Extending this perspective to the reweighted case, RWL1 can be seen as a MAP estimator using a Gaussian likelihood and a *heirarchical* prior consisting of a Laplacian prior and Gamma hyperprior. (The Gamma distribution is conjugate to the Laplacian distribution, facilitating analytic computation of necessary quantities). Introducing the heirarchical prior allows prior knowledge to by incorporated into the hyperparameters governing the shape of the Gamma hyperprior; injecting prior knowledge into the second order statistics in this way is a particularly robust method for incorporating prior knowledge.

Inspired by [6], [3] uses this key insight to introduce a highly robust dynamic filtering algorithm, RWL1-DF, which uses prior estimates of a slowly changing signal to improve estimates made from streaming observations. Mathematically, incorporating these estimates into the hyperparameters of the next time step is akin to incorporating them into the weights with some additive and multiplicative offsets. Again, the key feature of this algorithm is the robustness that comes from the sparsity-encouraging properties of the Laplacian prior coupled with the incorporation of prior knowledge through second-order statistics. In particular (and in contrast to support estimation based methods), a small previous estimate leads to large weights which encourage (but not force) the next estimate to be zero; a large previous estimate leads to small weights which encourage (but not force) the next estimate to be nonzero.

In addition to its robustness, the RWL1-DF strategy is also desirable for its simplicity. Therefore, we plan to use it in our project to extend the LVS-glasso (ref equation number) to the dynamic case. In particular, we can incorporate an estimate of edge weights in the Graphical model obtained from a previous time step by into the LVS-glasso objective by adding a weights to the sparsity-inducing penalty on the sparse connections $\boldsymbol{S}$, $\|\boldsymbol{S}\|_1$. We also plan to experiment with different methods for weighting (and therefore tracking) the low rank component $\boldsymbol{L}$ that governs the connections between the observed and hidden variables in the GGM such as weighting rows and columns differently or weighting each singular value (following work on tracking with nuclear norm minimization).

Prelimary Results We have started implementing the baseline Gaussian graphical model learning algorithms. After creating a 50-node sparse Gaussian graphical model as specified in [2] (a grid of points in a 1x1 box where edges are added with probability proportional to distance between points), we ran the latent-variable glasso method and inferred the below graphs, achieving almost an exact match in graphs. After applying SVD decomposition on the low-rank matrix L, we noticed that first two singular values are the primary components of the matrix, while the remaining fall off drastically. We will continue to expand our results to compare LV-glasso with glasso, and also apply the methods on actual datasets as described above.





Implementation plan and datasets. The first stage of our project will involve implementing the static algorithm in [2] and static time series algorithm in [5] in MATLAB. We plan to implement the algorithmic components in software ourselves, using convex optimization Next, we will implement and evaluate several approaches to incorporating temporal dynamics using two datasets:

1. *Stock data*: First, we plan to demonstrate our dynamic algorithm to recover a GGM for time series data of major global stock indices. This will allow us to compare the results of our temporal-dynamics aware algorithm with the similar experiments for the non temporal dynamics aware algorithm in [4]. We have already collected a 3-year dataset from 13 major global stock indices using data available on Google finance.
2. *Symptom network analysis for major depressive disorder*: Symptom networks are used to visualize relationships between symptoms in order to develop informed treatment plans based on measures of centrality (possible driving symptoms). Learning sparse graphical models from time-series survey responses with dynamic symptom presentation could be a high impact application of our algorithm [1].