The joint graphical lasso for inverse covariance estimation across multiple classes

Ariane Stark and Margaret Janiczek

Section 1 to Section 3

Margaret’s Overview

Section 4: Faster computations for the FGL and GGL

Computational Time Improvements

Empirical covariance matrices $S^{(1)} \cdots S^{(K)}$ can be permuted so that the output of the JGL algorithm is block diagonal. The JGL algorithm can be performed on the blocks separately producing the exact same result as the result on all p features.

If for a given choice λ_1 and λ_2 the estimated inverse covariance matrices $\hat{\Theta}^{(1)} \cdots \hat{\Theta}^{(K)}$ are block diagonal each with the same R blocks. Let the number of features for the rth block be denoted p_r where $\sum_{r=1}^{R} p_r = p$.

Theorems ensure that this block structure is possible. It is important to note that the improvements in speed only exist if λ_1 and λ_2 are sufficiently large.

Section 5: Relationship to previous proposals

Guo et. al (2011) proposal and relation to authors methods

The proposal by Guo et al. (2011) is closest to the authors' method with a hierarchical group lasso penalty that encourages a shared pattern of sparsity across the K classes.

$$P(\{\Theta\}) = \lambda \sum_{i \neq j} \left(\sum_{k} |\theta_{ij}^{(k)}| \right)^{1/2}$$

Some disadvantages of the proposal by Guo et al compared ot FGL and GGL

- This penalty is not convex, so algorithmic convergence to the global optimum is not guaranteed. Therefore,it is not possible to achieve the improvements in speed.
- The approach is quite slow relative to the authors approach and essentially cannot be applied to very high dimensional data sets.
- It uses just one tuning parameter and cannot control separately the sparsity level and the extent of network similarity.
- In cases where one expects edge values as well as network structure to be similar between classes Guo et al. (2011) encourage shared patterns of sparsity but do not encourage similarity in the signs and values of the non-zero edges.

Section 6: Tuning Parameter Selection

Slide 1

The authors recommend an "application-driven selection of tuning parameters to achieve a model that is biologically plausible, sufficiently complex to be interesting and sufficiently sparse to be interpretable and extremely well supported by the data. In fact, network estimation methods will often prove most descriptively useful when run over a variety of tuning parameters, giving the researcher a sense of how easily various edges overcome increasing values of the sparsity penalty and how readily they become shared across networks as the similarity penalty increases. Ideally, the final model would be accompanied by a p-value on each edge or an overall estimate of the edge false discovery rate (FDR), a difficult problem that was addressed by Li et al. (2013) in the partial correlation-based network estimation framework and an important goal for research in likelihood-based network estimation methods."

AIC Approximation

Selection of tuning parameters for the JGL by using an approximation of the AIC through a grid search can then be performed to select λ_1 and λ_2 that minimize the $\mathrm{AIC}(\lambda_1, \lambda_2)$ score:

$$\mathrm{AIC}(\lambda_1, \lambda_2) = \sum_{k=1}^{K} [n_k \mathrm{tr}(S^{(k)} \hat{\Theta}^{(k)}_{\lambda_1, \lambda_2}) - n_k \log\{\det(\hat{\Theta}^{(k)}_{\lambda_1, \lambda_2})\} + 2E_k]$$

- $\hat{\Theta}^{(k)}_{\lambda_1, \lambda_2}$ inverse covariance matrix
- E_k the number of non-zero elements in $\hat{\Theta}^{(k)}_{\lambda_1, \lambda_2}$

When the number of variables p is very large a dense search over λ_1 while holding λ_2 at a fixed low value, followed by a quick search over λ_2, holding λ_1 at the selected value is suggested.

Section 7: Simulation Study

Overview

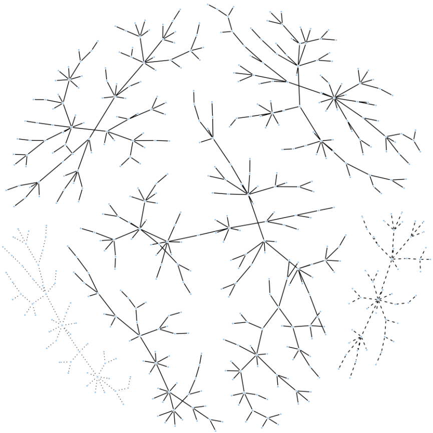
The authors compare the performances of the FGL and GGL with two existing methods, the graphical lasso and the proposal of Guo et al. (2011). When applying the graphical lasso, networks are fitted for each class separately. They also go on to investigate the effects of n and p on the FGL and GGL's performances.

The effects of the FGL and GGL penalties vary with the sample size so, for ease of presentation of the simulation study results, λ_1 and λ_2 are multiplied by the sample size of each class before performing the JGL.

To ease interpretation, the GGL penalties were reparameterized in the simulation study in terms of 'sparsity'(ω_1) and 'similarity'(ω_2). In the FGL, to reparameterization is needed for λ_1 and λ_2, as the former drives network sparsity and the latter drives network similarity.

Performance as a function of tuning parameters

This simulation considers a three-class problem of three generated networks with $p = 500$ features belonging to 10 equally sized unconnected subnetworks, each with a distribution thought to mimic the structure of biological networks. Of the 10 subnetworks, eight have the same structure and edge values in all three classes, one is identical between the first two classes and missing



Network used to generate simulated data sets: full edges are common to all three classes; broken edges are present only in classes 1 and 2, and dotted edges are present only in class 1.

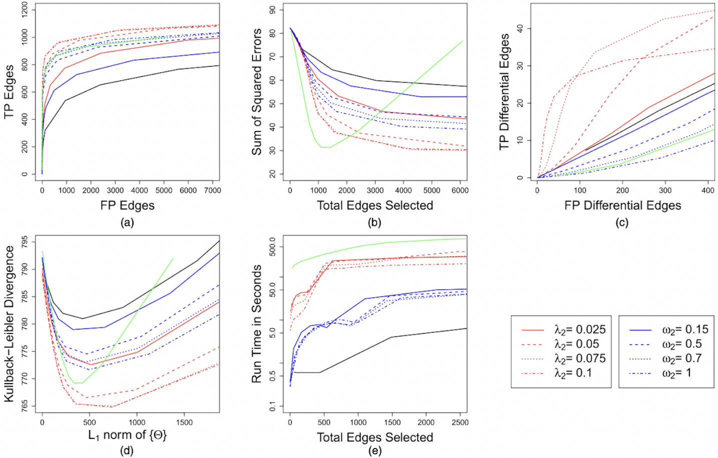


Fig. 2. Performance of the FGL (——), GGL (——), the method of Guo *et al*. (2011) (——) and the graphical lasso (——) on simulated data with 150 observations in each of three classes and 500 features: (a) number of edges correctly identified to be non-zero (true positive edges) *versus* number of edges incorrectly identified to be non-zero (false positive edges); (b) sum of squared errors in edge values *versus* the total number of edges estimated to be non-zero; (c) number of edges correctly found to have values differing between classes (true positive differential edges) *versus* the number of edges incorrectly found to have values differing between classes (false positive differential edges); (d) dKL for the estimated models from the true models *versus* the l_1-norm of the off-diagonal entries of the estimated precision matrices; (e) running time *versus* the number of non-zero edges estimated (note the use of a log-scale on the vertical axis)

Performance as a function of tuning parameters: Results

- A: As the sparsity tuning parameters decrease,the number of edges selected increases. The FGL, GGL and the proposal of Guo et al. (2011) dominate the graphical lasso.
- B: FGL, GGL and the graphical lasso tend to overshrink edge values towards zero owing to the use of convex penalty functions. The FGL and GGL attain sum of squared errors values that are as low as those of Guo et al. (2011) but do so when estimating much larger networks.
- C: The FGL yields fewer false positive results than the competing methods, since it shrinks between-class differences in edge values to 0. Neither the GGL nor the method of Guo et al. (2011) is designed to shrink edge values towards each other so neither method outperforms even the graphical lasso.

- D: At most values of λ_2, the FGL attains a lower Kullback–Leibler divergence (dKL) than the other methods, followed by the method of Guo et al. (2011) and then by the GGL. The graphical lasso has the worst performance, since it estimates each network separately.
- E: The graphical lasso is fastest, but the FGL and GGL are much faster than the proposal of Guo et al. (2011). Note the FGL algorithm is much faster in problems with only two classes, since in that case there is a closed form solution to the generalized fused lasso problem

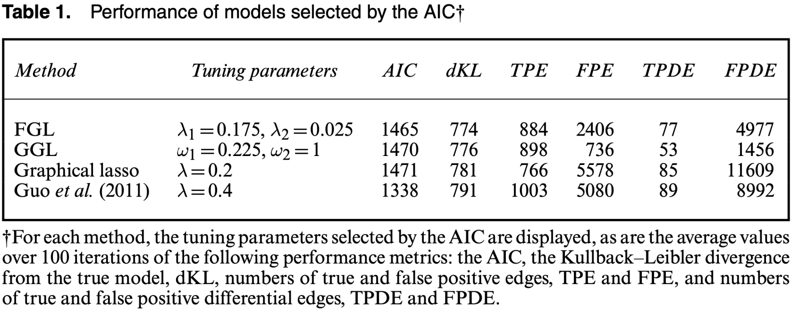


Table 1. Performance of models selected by the AIC†

Method	*Tuning parameters*	*AIC*	*dKL*	*TPE*	*FPE*	*TPDE*	*FPDE*
FGL	$\lambda_1=0.175$, $\lambda_2=0.025$	1465	774	884	2406	77	4977
GGL	$\omega_1=0.225$, $\omega_2=1$	1470	776	898	736	53	1456
Graphical lasso	$\lambda=0.2$	1471	781	766	5578	85	11609
Guo *et al.* (2011)	$\lambda=0.4$	1338	791	1003	5080	89	8992

†For each method, the tuning parameters selected by the AIC are displayed, as are the average values over 100 iterations of the following performance metrics: the AIC, the Kullback–Leibler divergence from the true model, dKL, numbers of true and false positive edges, TPE and FPE, and numbers of true and false positive differential edges, TPDE and FPDE.

The AIC-selected FGL and GGL models outperform the AIC-selected models from the earlier methods. The AIC selects a larger model for the method of Guo et al.than it does for the FGL and GGL. For all methods, the AIC appears to select models with low dKL from the truth but with greater numbers of edges than would be ideal for accurate hypothesis generation. The AIC selected a much smaller model for the GGL than for the other methods, achieving by far the best edge estimation performance.

Performance as a function of n and p

A network was generated with $p = 500$ with similar set up as the prior simulation using K = 2 instead of K = 3. The first network has 10 equal-sized components with power law degree distributions (distribution thought to mimic the structure of biological networks). The second network is identical to the first in both edge identity and value, but with two components removed. In addition to the 500-feature network pair, a pair of networks with $p = 1000$ features was generated, each of which is block diagonal with 500 × 500 blocks corresponding to two copies of the 500-feature networks.

For both the $p = 500$ and the $p = 1000$ networks, 100 data sets were simulated with $n = 50$, $n = 200$ and $n = 500$ samples in each class. FGL and GGL were run with λ_1 =0.2 and λ_2 =0.1, λ_1 =0.05 and $\lambda_2 = 0.2$ respectively.

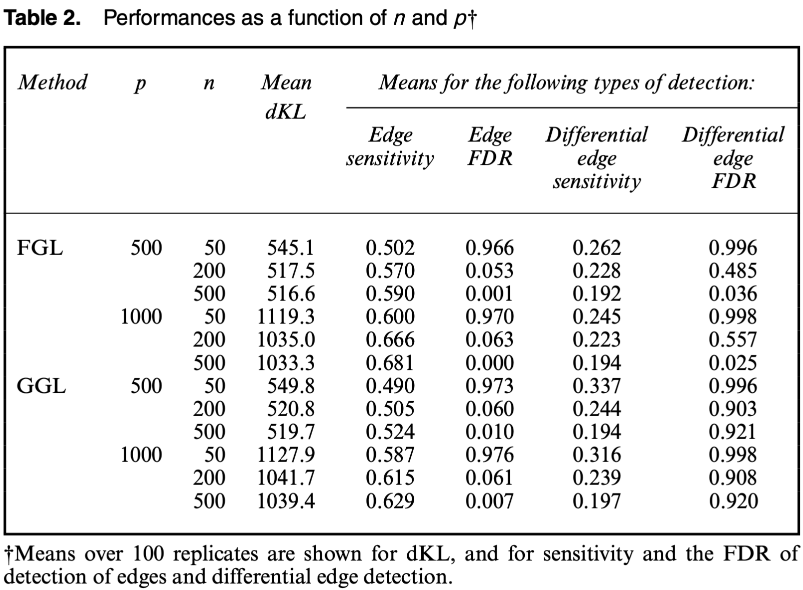


Table 2. Performances as a function of n and p†

Method	*p*	*n*	*Mean dKL*	*Means for the following types of detection:*			
				Edge sensitivity	*Edge FDR*	*Differential edge sensitivity*	*Differential edge FDR*
FGL	500	50	545.1	0.502	0.966	0.262	0.996
		200	517.5	0.570	0.053	0.228	0.485
		500	516.6	0.590	0.001	0.192	0.036
	1000	50	1119.3	0.600	0.970	0.245	0.998
		200	1035.0	0.666	0.063	0.223	0.557
		500	1033.3	0.681	0.000	0.194	0.025
GGL	500	50	549.8	0.490	0.973	0.337	0.996
		200	520.8	0.505	0.060	0.244	0.903
		500	519.7	0.524	0.010	0.194	0.921
	1000	50	1127.9	0.587	0.976	0.316	0.998
		200	1041.7	0.615	0.061	0.239	0.908
		500	1039.4	0.629	0.007	0.197	0.920

†Means over 100 replicates are shown for dKL, and for sensitivity and the FDR of detection of edges and differential edge detection.

The accuracy of covariance estimation (dKL) improves significantly from n = 50 to n = 200, and it improves only marginally with a further increase to n = 500. Detection of edges improves throughout the range of ns sampled: for both the FGL and the GGL, sensitivity improves slightly with increased sample size, and the FDR decreases dramatically. Accurate detection of edge differences is a more difficult goal, though the FGL succeeds in it at higher sample sizes.

Section 8: Analysis of lung cancer microarray data

Data and λ_1

- 22283 microarray-derived gene expression measurements from large airway epithelial cells sampled from 97 patients with lung cancer and 90 controls (Spira et al., 2007).
- Omitted genes with standard deviations in the bottom 20% since a greater share of their variance is probably attributable to non-biological noise.
- Remaining genes were standardized to have mean 0 and standard deviation 1 within each class.
- Weighted each class equally instead of by sample size to avoid disparate levels of sparsity between the classes and to prevent the larger class from dominating the estimated networks
- Chose a high value for the sparsity tuning parameter, $\lambda_1 = 0.95$, to yield very sparse network estimates.

Results

- Ran the FGL with a range of λ_2 values to identify the edges that differed most strongly, and settled on $\lambda_2 = 0\text{-}005$ as providing the most interpretable results
- Only 278 genes were connected to any other gene by using the chosen tuning parameters
- Identification of block diagonal structure took less than 2 min.
- FGL estimated 134 edges shared between the two networks, 202 edges present only in the cancer network and 18 edges present only in the normal tissue network.

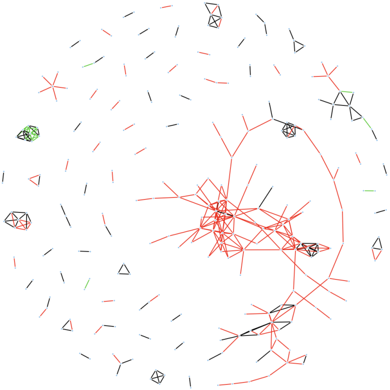


Fig. 3. Conditional dependence networks inferred from 17772 genes in normal and cancerous lung cells (278 genes have non-zero edges in at least one of the two networks): ——, edges common to both classes; ——, tumour-specific edges; ——, normal-specific edges

- The estimated networks contain many two-gene subnetworks that are common to both classes, a few small subnetworks and one large subnetwork specific to tumour cells.
- 45% of edges, including almost all of the two-gene subnetworks, connect multiple probes for the same gene. Many other edges connect genes that are obviously related, that are involved in the same biological process or that even code for components of the same enzyme.
- Recovery of these pairs suggests that the FGL (and other network analysis tools) can generate high quality hypotheses about gene coregulation and functional interactions.
- Increases confidence that some of the non-obvious two-gene subnetworks that are detected in this analysis may merit further investigation.

Future Studies

The small black and green network suggests an interesting phenomenon. It contains multiple probes for two haemoglobin genes. In the normal tissue network, the probes for these genes are heavily interconnected both within and between the genes. In the tumour cells, although edges between the genes are preserved, no edges connect the two genes. The abundance of connections between the two genes in healthy cells and the absence of connections in tumour cells may indicate a possible direction of future investigation.

Section 9: Discussion

- Algorithm is tractable on very large data sets (more than 20000 features) and usually converges in seconds for smaller problems (500 features).
- Methods outperform competing approaches over a range of simulated data sets.
- In the JGL optimization problem , the contribution of each class to the penalized log-likelihood is weighted by its size. By omitting the n_k term it is possible to weight the classes equally to prevent a single class from dominating estimation.
- FGL and GGL's reliance on two tuning parameters is a strength rather than a drawback
- JGL has potential applications beyond those discussed in this paper

Metabolomics Data

Back to Margaret's Tutorial