LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Modeling under Dependence

Graphical lasso and Mixed Graphical Model

Patricio Massaro[1]
p.massaro@campus.lmu.de

Dusan Urosevic[1]
dusan.urosevic@campus.lmu.de

[1]LMU München

May 9, 2024

The problem

We have observational data $X_1, X_2, ..., X_n$ and we wish to derive the distribution $P(X)$.
Problem: The dependence structure of the variables is unknown and/or can't be estimated using standard ML approach due to $P >> N$.

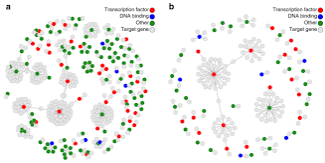


Figure 1: Gene networks

Figure 2: Car traffic

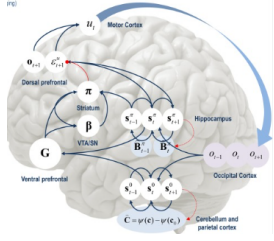
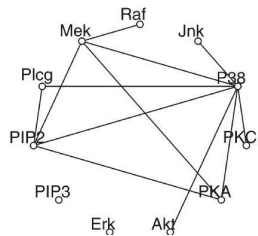


Figure 3: Brain activity

Gaussian Markov Fields - Recap

- **Markov property** of graphical model - Lack of edge X_i, X_j implies conditional independence $X_i \perp X_j \mid X \setminus \{X_i, X_j\}$ i.e. the dependence is localised
- $X_1, X_2, ..., X_n$ come from a MV Gaussian distribution
- **Result:** Determining precision matrix $\Theta\left(\Sigma^{-1}\right)$ is equivalent to structure learning

$$\Theta_{i,j} = 0 \Leftrightarrow X_i \perp X_j \mid X \setminus \{X_i, X_j\}$$



Heuristic - L1 penalization

MV Gaussian

$$X \overset{d}{\sim} \mathcal{N}(x|\mu, \Sigma),\ p(X) = \frac{1}{(2\pi)^{(p/2)}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^{\mathsf{T}}\,\Sigma^{-1}\,(x-\mu)\right\}$$

$$= \frac{1}{(2\pi)^{(p/2)}}|\Theta|^{1/2} \exp\left\{-\frac{1}{2}(x-\mu)^{\mathsf{T}}\,\Theta\,(x-\mu)\right\}$$

Likelihood

$$\ell(x;\mu,\Sigma) \propto \frac{n}{2}\log\det(\Theta) - \frac{1}{2}\sum_{i=1}^{n}(x^{(i)}-\mu)^{\mathsf{T}}\Theta(x^{(i)}-\mu)$$

$$\ell(x;\mu,\Sigma) \propto \frac{n}{2}\log\det(\Theta) - \frac{n}{2}\mathrm{tr}(S\Theta)$$

GLasso - Definition

Assumption: GMRF with a sparse structure (few non-zero values in Θ)

- Like in LASSO we introduce an L1 penalty to the ML objective

$$\log\det\Theta - \text{tr}(S\Theta) - \lambda\|\Theta\|_1$$

- where S is the empirical covariance matrix and λ is the penalisation parameter.
- $\|\Theta\|_1 = \sum_{i=1}^{n}\sum_{j=1}^{n} |\Theta_{i,j}|$, $\text{tr}(A) = \sum_{i=1}^{n} a_{ii}$

Pair-wise regression

Perform **Lasso** for each variable independently, producing:

$$x_1 = \boxed{c_{12}} x_2 + c_{13}x_3 + \ldots + c_{1p}x_p$$

$$x_2 = \boxed{c_{21}} x_1 + c_{23}x_3 + \ldots + c_{2p}x_p$$

$$\ldots$$

$$x_p = c_{p1}x_1 + c_{p2}x_2 + \ldots + c_{p(p-1)}x_{(p-1)}$$

Structure learning: $c_{ij} = 0$ and $c_{ji} = 0 \implies X_i \perp X_j \mid X \setminus \{X_i, X_j\}$

Issue: This approximation is consistent in recovering the structure when $n \to \infty$ it has **no guarantees** on convergence speed

GLasso - Method

- Suppose we have N multivariate normal observations of dimension p. The distribution is specified by (Σ, μ). With $\hat{\Sigma}$ we denote our estimate of Σ. Using the partitioning:

$$\hat{\Sigma} = \begin{pmatrix} \hat{\Sigma}_{11} & \hat{\sigma}_{12} \\ \hat{\sigma}_{12}^T & \hat{\sigma}_{22} \end{pmatrix}, \quad S = \begin{pmatrix} S_{11} & s_{12} \\ s_{12}^T & s_{22} \end{pmatrix}$$

- The maximum of the **L1** penalised log-likelihood over Θ satisfies:

$$\hat{\sigma}_{12} = \text{argmin}_y \left\{ y^T \hat{\Sigma}_{11}^{-1} y : \|y - s_{12}\|_\infty \leqslant \lambda \right\}, \text{similar to MV regression coefficients}$$

GLasso - Method

$$\hat{\sigma}_{12} = \text{argmin}_y \left\{ y^T \hat{\Sigma}_{11}^{-1} y : \|y - s_{12}\|_\infty \leqslant \lambda \right\}$$

Main idea:

- This is a box-constrained quadratic program (QP).
- Block coordinate descent. In each iteration update only one row and column of $\hat{\Sigma}$ (rotated so it is always last) and solve the above problem. Repeat until convergence.
- Key advantage: If $\hat{\Sigma}$ is initialised with a PD matrix solution will exist even if $p > N$.

GLasso - Dual problem

Using convex duality, *Banerjee and others (2007)* go on to show that solving $\hat{\sigma}_{12} = \text{argmin}_y \left\{ y^T \hat{\Sigma}_{11}^{-1} y : \|y - s_{12}\|_\infty \leqslant \lambda \right\}$ is equivalent to solving the dual problem:

$$\min_\beta \left\{ \frac{1}{2} \left\| \hat{\Sigma}_{11}^{1/2} \beta - b \right\|^2 + \lambda \|\beta\|_1 \right\},$$

where $b = \hat{\Sigma}_{11}^{-1/2} s_{12}$. If β solves the dual problem then $\hat{\sigma}_{12} = \hat{\Sigma}_{11} \beta$ solves the primal problem. Also we have that

$$\theta_{12} = -\theta_{22} \beta, \text{ where, } \Theta = \begin{pmatrix} \Theta_{11} & \theta_{12} \\ \theta_{12}^T & \theta_{22} \end{pmatrix}$$

GLasso - Algorithm

1. Start with $\hat{\Sigma} = S + \lambda I$. The diagonal of $\hat{\Sigma}$ remains unchanged in what follows.
2. For each $j = 1, 2, \ldots p$, solve the lasso problem $\min_\beta \left\{ \frac{1}{2} \left\| \hat{\Sigma}_{11}^{1/2} \beta - b \right\|^2 + \lambda \|\beta\|_1 \right\}$, which takes as input the inner products $\hat{\Sigma}_{11}$ and s_{12}. This gives a $p-1$ vector solution $\hat{\beta}$. Fill in the corresponding row and column of $\hat{\Sigma}$ using $\hat{\sigma}_{12} = \hat{\Sigma}_{11}\hat{\beta}$
   - If $avg(\Delta\hat{\sigma}_{ij}) < t$ stop algorithm
3. Repeat step **2.**

Note: Pair-Wise Regression solves the lasso problems independently, **GLasso** performs joint optimisation. Information sharing is achieved through the shared $\hat{\Sigma}$.

GLasso - Update step

In each step we permutate the rows and columns so that the target variable is last. We solve a single lasso problem using coordinate descent. Let $V = \hat{\Sigma}_{11}$, and $u = s_{12}$. Then, we have that for $j = 1, 2, \ldots p$:

$$\hat{\beta}_j \leftarrow \frac{S\left(u_j - \sum_{k \neq j} V_{kj}\hat{\beta}_k, \lambda\right)}{V_{jj}}$$

where S is the **soft-threshold** operator:

$$S(x, \lambda) = \text{sign}(x)(|x| - \lambda)_+$$

Recovering the Precision Matrix

Although our algorithm has calculated $\hat{\Sigma}$, we can recover $\hat{\Theta} = \hat{\Sigma}^{-1}$. Using

$$\hat{\Sigma}_{11}\theta_{12} + \hat{\sigma}_{12}\theta_{22} = 0$$
$$\hat{\sigma}_{12}^T\theta_{12} + \hat{\sigma}_{22}\theta_{22} = 1$$
$$\hat{\beta} = \hat{\Sigma}_{11}^{-1}\hat{\sigma}_{12}$$

we derive:

$$\hat{\theta}_{22} = 1/\left(\hat{\sigma}_{22} - \hat{\sigma}_{12}^T\hat{\Sigma}_{11}^{-1}\hat{\sigma}_{12}\right) = 1/\left(\hat{\sigma}_{22} - \hat{\sigma}_{12}^T\hat{\beta}\right)$$
$$\hat{\theta}_{12} = -\hat{\Sigma}_{11}^{-1}\hat{\sigma}_{12}\hat{\theta_{22}} = -\hat{\beta}\hat{\theta}_{22}$$

Note: $\hat{\theta}_{22}$ doesn't change

GLasso - Use Case

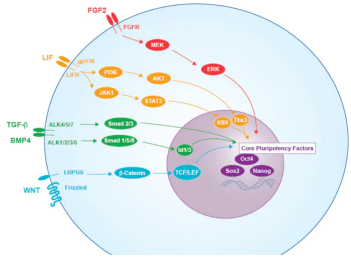
Determining Protein-Signaling networks

Data:

- Measurements of concentraition of 11 signaling proteins
- 7500 individual cells

Goal: Determine **dependence structure** between these proteins

We make use of the GLasso implementation in R provided by Friedman.



GLasso - Results

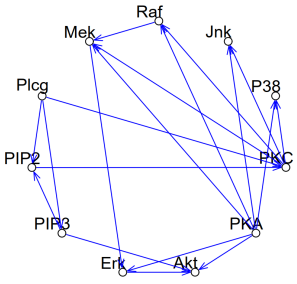


Figure 4: Ground truth

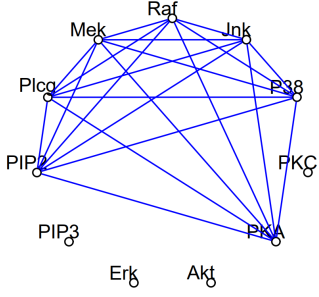


Figure 5: GLasso - Friedman

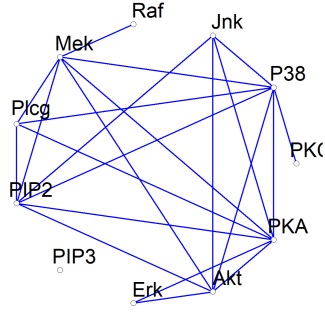
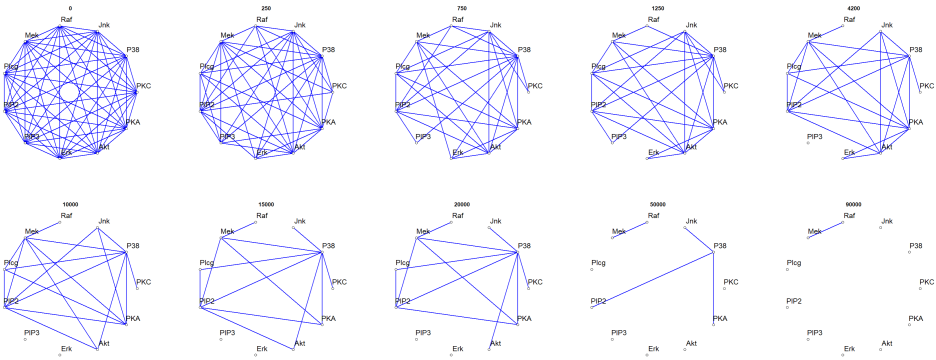
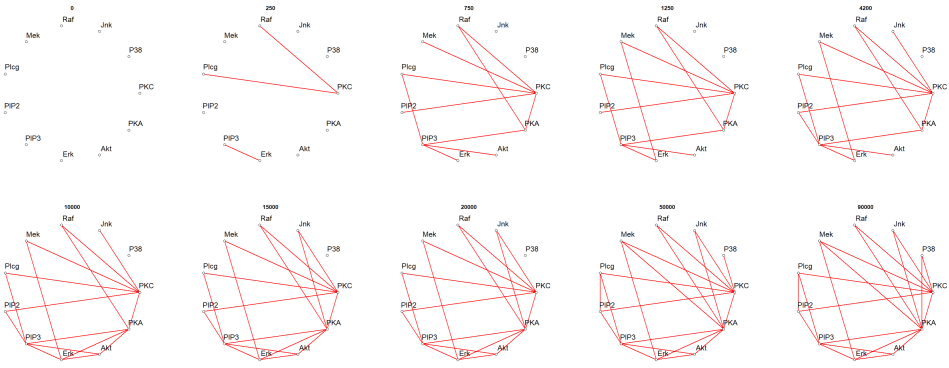


Figure 6: Our result

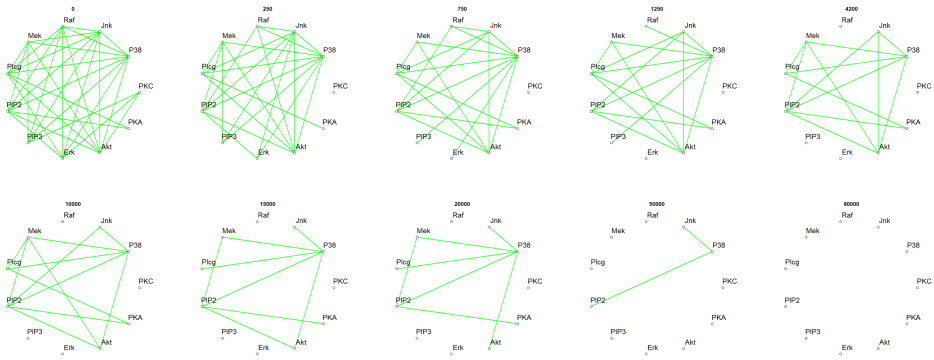
GLasso - Varying λ



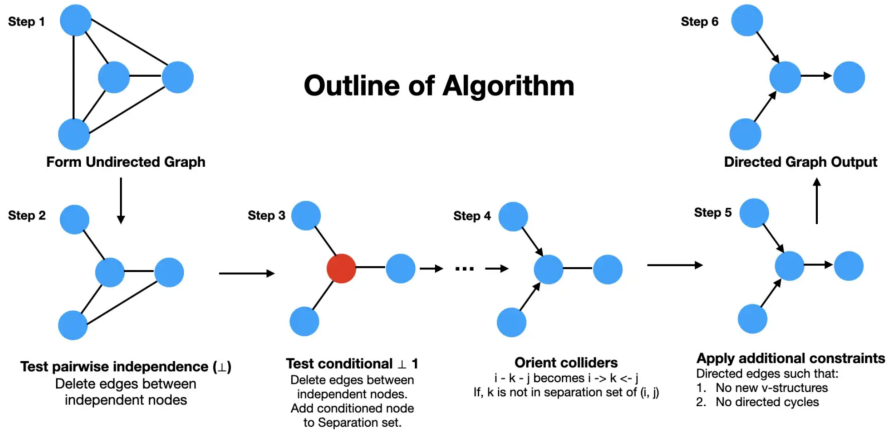
GLasso - Lost edges



GLasso - False edges



Quick Recap


Step 1
Form Undirected Graph
Outline of Algorithm
Step 6
Directed Graph Output
Step 2
Test pairwise independence (⊥)
Delete edges between independent nodes
Step 3
Test conditional ⊥ 1
Delete edges between independent nodes.
Add conditioned node to Separation set.
Step 4
Orient colliders
i - k - j becomes i -> k <- j
If, k is not in separation set of (i, j)
Step 5
Apply additional constraints
Directed edges such that:
1. No new v-structures
2. No directed cycles

Heuristics - Comparison

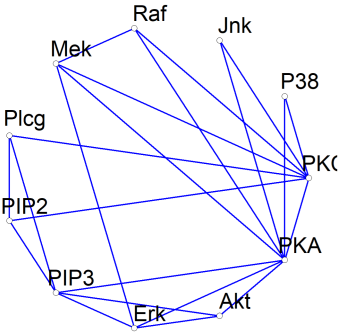


Figure 7: Ground truth

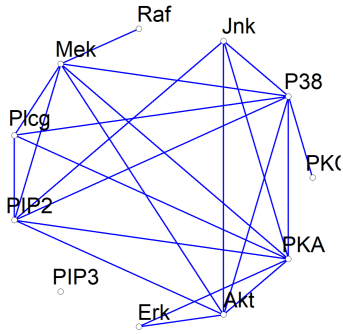


Figure 8: GLasso $\lambda = 4200$

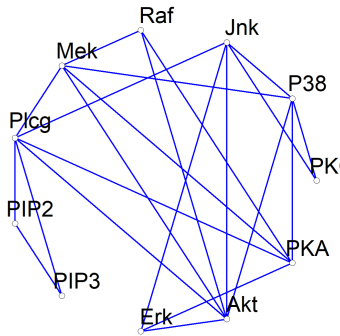


Figure 9: PC - $\alpha = 0.0005$

Heuristics - Comparison

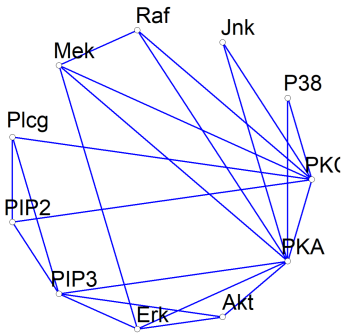


Figure 10: Ground truth

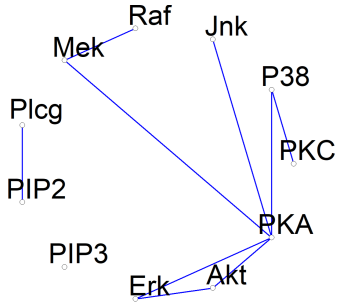


Figure 11: GLasso $\lambda = 4200$

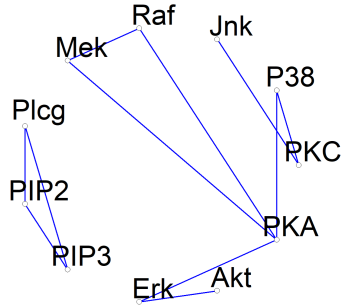


Figure 12: PC - $\alpha = 0.0005$

GLasso - Fattahi and Sojoudi 2017

Provides a **closed form** solution for GLasso that is :

- An exact solution in case of acyclic gprahs
- Approximate solution for arbitrary graphs

In addition, the authors further examine GLasso and provide insights about:

- Changes in sparsity caused by changes in λ
- Equivalence of GLasso with **Thresholding** S for very sparse graphs

GLasso - Fattahi and Sojoudi 2017

d	m	Closed-Form	QUIC-C	QUIC-W	GLASSO-C	GLASSO-W	Elem.
2000	9894	0.1	2.0	1.4	42.8	13.5	0.2
2000	20022	0.1	3.0	2.1	43.8	15.3	0.2
4000	20094	0.5	13.9	7.5	460.8	135.1	2.1
4000	40382	0.5	21.5	12.0	467.6	156.2	2.9
8000	40218	2.5	78.7	49.3	3675.1	1011.2	11.3
8000	79890	2.5	111.7	88.4	3784.3	1278.8	22.2
12000	60192	7.8	243.8	153.1	⋆	3233.0	31.8
12000	119676	7.4	333.6	251.0	⋆	3437.2	70.2
16000	80064	17.1	570.0	322.8	⋆	6545.0	67.2
16000	160094	18.5	787.4	616.4	⋆	9960.8	174.8
20000	99954	39.4	1266.5	539.4	⋆	⋆	107.8
20000	200018	37.4	1683.8	1392.5	⋆	⋆	211.5
40000	200290	495.4	⋆	⋆	⋆	⋆	⋆
80000	401798	1450.4	⋆	⋆	⋆	⋆	⋆

Figure 13: Runtime comparison

GLasso - Fattahi and Sojoudi 2017

$$S_{ij}^{opt} = \begin{cases} \frac{1}{\Sigma_{ii}} \left(1 + \sum_{(i,m) \in \mathcal{E}^{opt}} \frac{(\Sigma_{im}^{\text{res}})^2}{\Sigma_{ii}\Sigma_{mm} - (\Sigma_{im}^{\text{res}})^2} \right) & if \quad i = j, \\ \frac{-\Sigma_{ij}^{\text{res}}}{\Sigma_{ii}\Sigma_{jj} - (\Sigma_{ij}^{\text{res}})^2} & if \quad (i,j) \in \mathcal{E}^{opt}, \\ 0 & otherwise, \end{cases}$$

Figure 14: Exact solution for acyclic graphs

GLasso - MCPeSe 2020

Issue:

- **Standard GLasso** is efficient but doesn't have a method for choosing the regularisation parameter λ.
- **Bayesian** implementations allow for priors of λ, but are inefficient in high-dimensional settings.

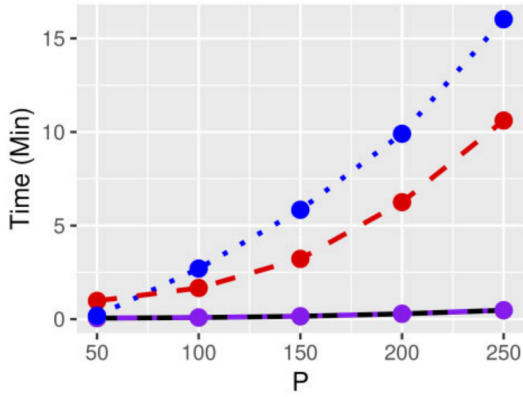
Solution: Use a Monte Carlo approach

- With Rejection Sampling/Metropolis-Hastings we directly simulate the posterior of λ
- Better runtime and similar performance in structure learning

MCPeSe - Runtime comparison

Legend:

- **Black** - MCPeSe
- **Purple** - RIC - Rotation Information Criterion
- **Red** - StARS - Stability Regularisation selection
- **Blue** - RIC - Bayesian GLasso



Intro

Key points:

1. Allows for learning of a Graphical model from data with both continuous and discrete variables
2. Based on a generalisation of edge penalisation from GLasso
3. Pair-wise MRF

$$p(y) \propto \exp \sum_{r \leq j} \phi_{rj}(y_r, y_j)$$

4. Pseudo-Likelihood (PL) based approach

Model: Pairwise MRF with density

$$p(x, y; \Omega) \propto \exp\left(\sum_{s=1}^{p}\sum_{t=1}^{p} -\frac{1}{2}\beta_{st}x_s x_t + \sum_{s=1}^{p}\alpha_s x_s + \sum_{s=1}^{p}\sum_{j=1}^{q}\rho_{sj}(y_j)\,x_s + \sum_{j=1}^{q}\sum_{r=1}^{q}\phi_{rj}(y_r, y_j)\right)$$

- x_s is the s-th out of p continuous variables, y_j is the j-th out of q discrete variables
- **Model parameters**: $\Omega = [\{\beta_{st}\}, \{\alpha_s\}, \{\rho_{sj}\}, \{\phi_{rj}\}]$ represent different types of node and edge potentials
- Note: ρ_{sj} - vector of length L_j, ϕ_{rj} - L_jxL_r matrix. Dummy encoding is used.

Features

The model simplifies to:

1. MV Gaussian model in case of only **continuous** data
2. Discrete pairwise MRF in case of only **discrete** data

The conditional distributions are:

1. Gaussian - For a single continuous variable conditioned on the rest
2. Multinomial - For a single discrete variable conditioned on the rest

- Note - using dummy coding for discrete variables

Maximum Likelihood:

Issue: When estimating Ω using MLE we get:

$$\log p(x, y; \Omega) = \sum_{s=1}^{p} \sum_{t=1}^{p} -\frac{1}{2} \beta_{st} x_s x_t + \sum_{s=1}^{p} \alpha_s x_s + \sum_{s=1}^{p} \sum_{j=1}^{q} \rho_{sj}(y_j) x_s + \sum_{j=1}^{q} \sum_{r=1}^{j} \phi_{rj}(y_r, y_j) - \log Z(\Omega)$$

Although this is a **convex** function using a Gradient descent based approach doesn't work.
Why? It involves the **NP-hard** subproblem of calculating the Partition function $Z(\Omega)$.

Solution: Use the pseudolikelihood

Pseudolikelihood

Originally introduced in Besag (1975) - computationally efficient and consistent estimator.

We replace the joint distribution $p(x, y; \Omega)$ with a product of all conditional distributions:

$$\tilde{\ell}(\Omega \mid x, y) = -\sum_{s=1}^{p} \log p\left(x_s \mid x_{|s}, y; \Omega\right) - \sum_{r=1}^{q} \log p\left(y_r \mid x, y_r; \Omega\right)$$

Pseudolikelihood

For a **continuous** variable x_s:

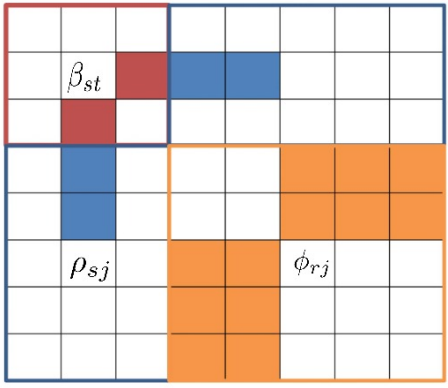
$$p\left(x_s \mid x_{\setminus s}, y; \Omega\right) = \frac{\sqrt{\beta_{ss}}}{\sqrt{2\pi}} \exp\left(\frac{-\beta_{ss}}{2}\left(\frac{\alpha_s + \sum_j \rho_{sj}\left(y_j\right) - \sum_{t \neq s} \beta_{st} x_t}{\beta_{ss}} - x_s\right)^2\right)$$

For a **discrete** variable y_r:

$$p\left(y_r \mid y_{\setminus r,}, x; \Omega\right) = \frac{\exp\left(\sum_s \rho_{sr}\left(y_r\right) x_s + \phi_{rr}\left(y_r, y_r\right) + \sum_{j \neq r} \phi_{rj}\left(y_r, y_j\right)\right)}{\sum_{l=1}^{L_r} \exp\left(\sum_s \rho_{sr}(l) x_s + \phi_{rr}(l, l) + \sum_{j \neq r} \phi_{rj}\left(l, y_j\right)\right)}$$

Lee and Haste (2012) show that this is a jointly convex problem

Structure learning



imgflip.com

Conditional independence:

- x_s and x_t if scalar $\beta_{st} = 0$.
- x_s and y_j if $\rho_{sj} = \vec{0}$ (for all L_j values of y_j)
- y_r and y_j if $\phi_{rj} = \vec{0}$ (for all $L_r \times L_j$ combinations of y_r and y_j)

Note: All parameter blocks appear exactly twice in the pseudolikelihoods

Penalisation

Adding a penalty like

$$\lambda \left(\sum_{s<t} \mathbb{I}\left[\beta_{st} \neq 0\right] + \sum_{sj} \mathbb{I}\left[\rho_{sj} \neq 0\right] + \sum_{r<j} \mathbb{I}\left[\phi_{rj} \neq 0\right] \right)$$

would make the objective function non-convex. Instead we use the following:

$$\min_{\Omega} \ell_\lambda(\Omega) = \ell(\Omega) + \lambda \left(\sum_{s=1}^{p} \sum_{t=1}^{s-1} |\beta_{st}| + \sum_{s=1}^{p} \sum_{j=1}^{q} \|\rho_{sj}\|_2 + \sum_{j=1}^{q} \sum_{r=1}^{j-1} \|\phi_{rj}\|_F \right)$$

Weighting scheme

In addition to different norms the Mixed Graphical Model introduces different weights for each parameter group Ω_g.

$$\lambda \left(\sum_{s=1}^{p} \sum_{t=1}^{s-1} w_{st} \left| \beta_{st} \right| + \sum_{s=1}^{p} \sum_{j=1}^{q} w_{sj} \left\| \rho_{sj} \right\|_2 + \sum_{j=1}^{q} \sum_{r=1}^{j-1} w_{rj} \left\| \phi_{rj} \right\|_F \right)$$

Friedman (2007) showed that Ω_g is non-zero at the optimum if $\|\partial \ell / \partial \Omega_g\| > \lambda w_g$.
Lee and Hastie recommend that w_g is chosen such that $w_g \propto \sqrt{E_{p_F} \|\partial \ell / \partial \Omega_g\|^2}$.

Objective Function

$$\ell(\Omega) + \lambda \left(\sum_{s=1}^{p} \sum_{t=1}^{s-1} w_{st} \left|\beta_{st}\right| + \sum_{s=1}^{p} \sum_{j=1}^{q} w_{sj} \left\|\rho_{sj}\right\|_2 + \sum_{j=1}^{q} \sum_{r=1}^{j-1} w_{rj} \left\|\phi_{rj}\right\|_F \right)$$

with weights

$$w_{st} = \sigma_s \sigma_t, \quad w_{sj} = \sigma_s \sqrt{\sum_a p_a \left(1 - p_a\right)}, \quad p_a = \Pr\left(y_r = a\right)$$

$$w_{rj} = \sqrt{\sum_a p_a \left(1 - p_a\right) \sum_b q_b \left(1 - q_b\right)}, \quad q_b = \Pr\left(y_j = b\right)$$

Optimisation procedure

We use the **Proximal gradient** method to optimise for the objective function. In general this method is used when:

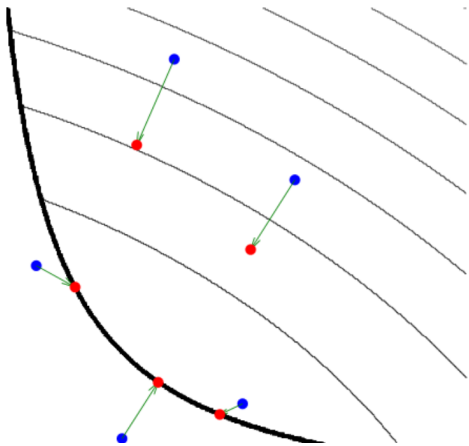
- The objective can be written as a sum of a Convex smooth function $f(x)$ and a Convex **non-smooth** function $g(x)$ (in our case this is the penalty term).
- We are able to calculate the derivative of $f(x)$ and the **proximal operator** for g(x):

Proximal operator

Motivation: For a convex function $g(x)$

$$\text{prox}_{gt}(x) = \underset{u}{\text{argmin}} \frac{1}{2t} \|x - u\|^2 + g(u)$$

1. Will move towards $g(x)$s optimum if x is in its domain
2. Will move onto the bondary of $g(x)$s domain and towards the optimum otherwise



Proximal gradient

Note: Minimising $f(x) + g(x)$ locally can be written as

$$\begin{aligned}&\underset{u}{\operatorname{argmin}} f(x_k) + \nabla f(x_k)^T (u - x_k) + \frac{1}{2t} \|u - x_k\|^2 + g(u) \\ &= \underset{u}{\operatorname{argmin}} \frac{1}{2t} \|u - (x_k - t\nabla f(x_k))\|^2 + g(u) = \operatorname{prox}_{gt}(x_k - t\nabla f(x_k))\end{aligned}$$

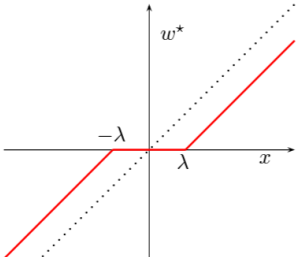
We use the following optimisation procedure:

1. $x'_k = x_k - t\nabla f(x_k)$ (GD on $f(x)$ - Pseudolikelihood)
2. $x_{k+1} = \operatorname{prox}_{gt}(x'_k)$. (PO of $g(x)$ - Penalty)
3. If convergence criteria not met return to step **1.**

Proximal Operator of the Mixed Model

For the Mixed Graphical Model the update step $x_{k+1} = \text{prox}_{gt}(x'_k)$ is equal to:

- Soft-thresholding for scalars β_{st}
- Group soft-thresholding on vectors ρ_{sj} and matrices ϕ_{rj}



$$S(x,t) = \left(1 - \frac{t}{|x|}\right)_+ x \tag{1}$$

$$S(\boldsymbol{w},t) = \left(1 - \frac{t}{||\boldsymbol{w}||}\right)_+ \boldsymbol{w} \tag{2}$$

Results

We used a survey dataset consisting of:

- 3000 samples
- $p = 2$ Continuous variables : Age and Logwage.
- $q = 6$ Categorical variables : Race, education level, marital status, health insurance, health, job class. 19 Total levels.
- $\beta \in \mathcal{R}^{2x2}$, $\rho \in \mathcal{R}^{2x19}$, $\phi \in \mathcal{R}^{19x19}$

Results

- If you want to take a look at the code, it's hosted in this github

```
Iteration:  250 objective:  495.5893
Iteration:  500 objective:  315.0845
Iteration:  750 objective:  234.9372
Iteration:  1000 objective:  194.9231
Iteration:  1250 objective:  172.2548
Iteration:  1500 objective:  157.6564
Iteration:  1750 objective:  148.6803
Iteration:  2000 objective:  144.7221
Iteration:  2250 objective:  142.9166
Converged!! Iteration:  2446 objective:  142.3672
```

Figure 16: A small step for the gradient, a giant leap for (our) humanity

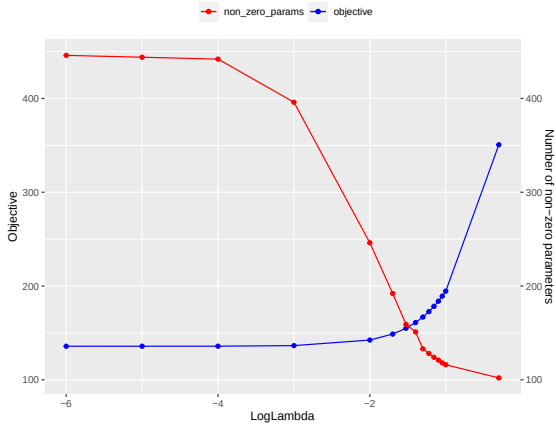


Figure 17: Objective and non-zero parameters

Mixed Graphical Model

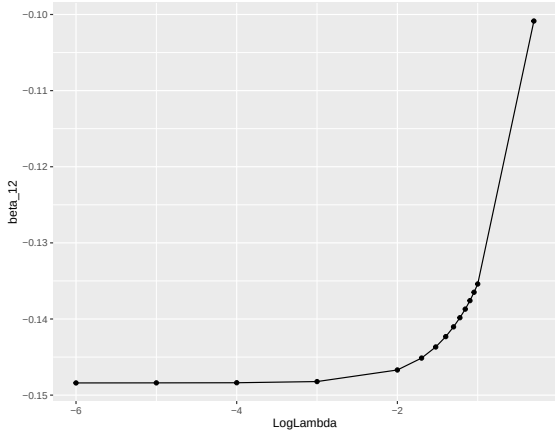


Figure 18: Interaction term between the two continuous variables

Mixed Graphical Model

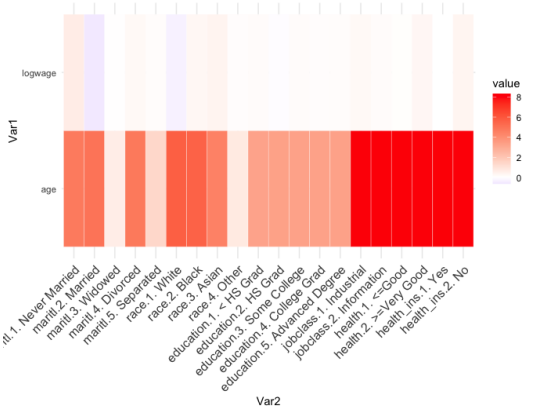


Figure 19: $\lambda = 10^{-5}$

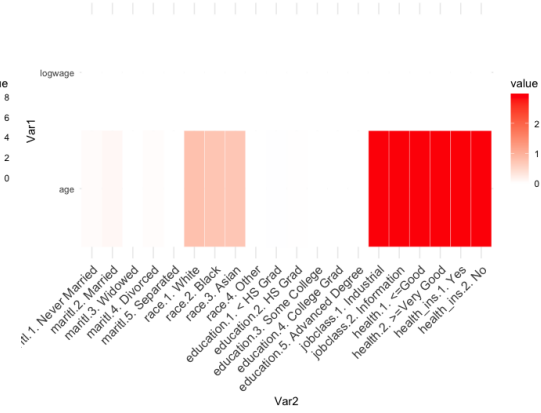


Figure 20: $\lambda = 1$

Mixed Graphical Model

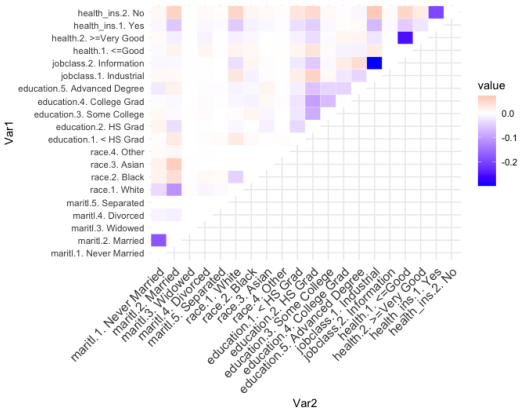


Figure 21: $\lambda = 10^{-5}$

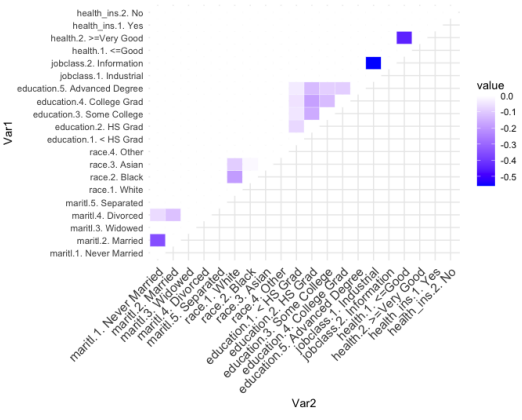


Figure 22: $\lambda = 1$

Mixed Graphical Model

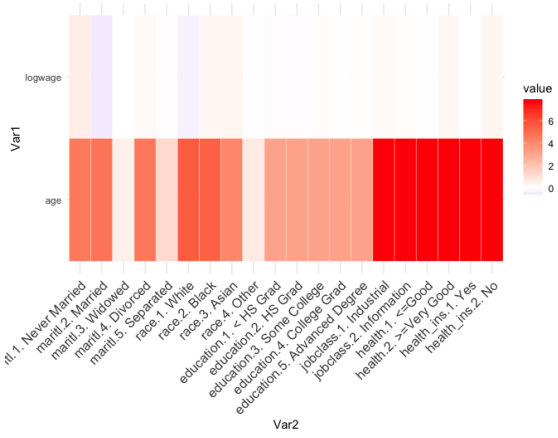


Figure 23: $\lambda = \sqrt{\frac{log(p+q)}{n}}$

Mixed Graphical Model

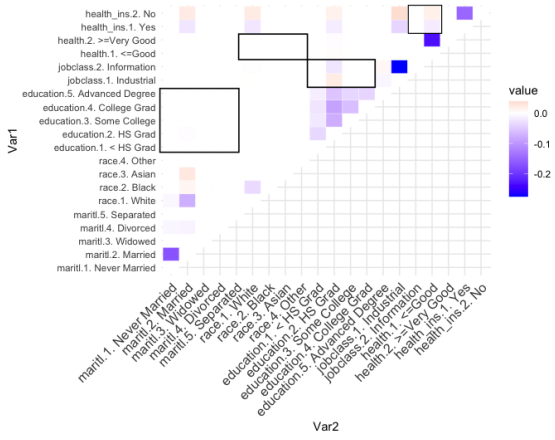


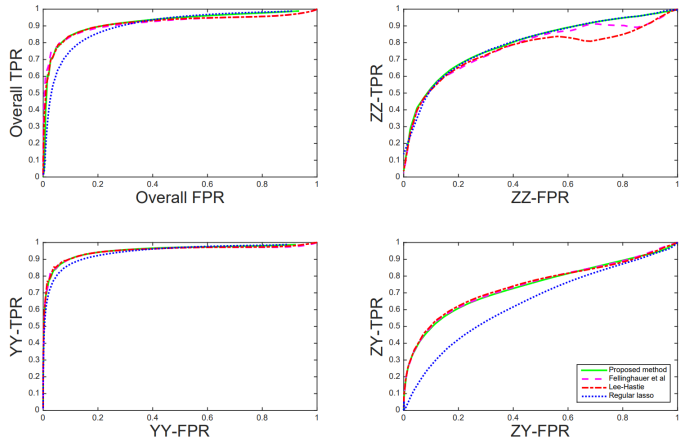
Figure 24: $\lambda = \sqrt{\frac{log(p+q)}{n}}$

MGM - High-Dimensional MGMs 2016

Key ideas:

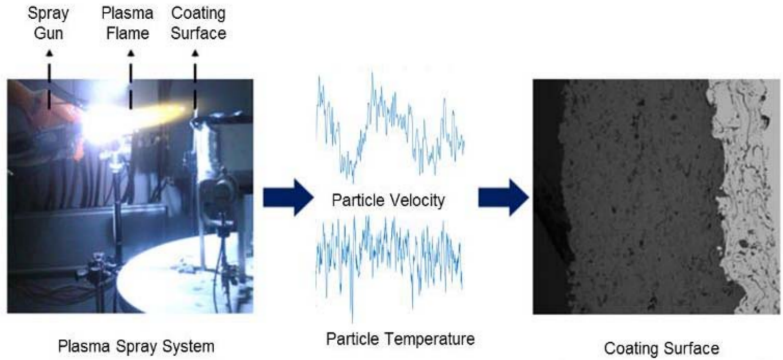
- Generalises original MGMs for High-dimensional data
- Flexible enough to capture any dependence structure - number of model parameters $O(max(q^2, p^2 q))$ instead of $O(p^2 2^{(p+q)})$ in the full Gaussian model
- Based on the conditional Gaussian density (PL approach)
- **Note:** Original MGMs used optimisation methods that optimise the PLs jointly. Here the parameters are obtained through Node-based regression (linear/logistic).

High-Dimensional MGMs - Results



MGM - Functional Graphical Models 2017

Example problem: Predicting results in industrial manufacturing.



MGM - Functional Graphical Models 2017

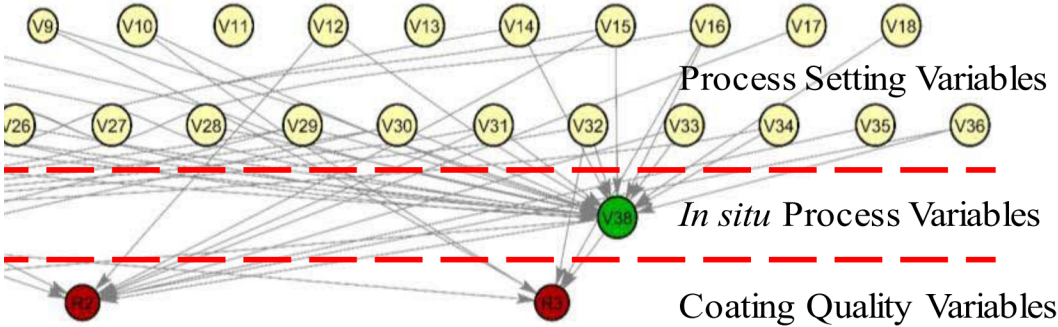
Before - Replace functional variable with summary static (ex.mean) and use L1 penalty

Functional Graphical Model:

- Represent function by its first m **basis functions** $(c_1, c_2, ..., c_m)$ (ex. FT)
- Use **separate penalisation** scheme for functional variables

$$\Omega_2(C) = \sum_{t=1}^{m} \| \boldsymbol{c}_t - \boldsymbol{c}_{\text{mean}} \|_2^2 = \| CR \|_F^2, R = \begin{bmatrix} 1 - \frac{1}{m} & -\frac{1}{m} & \cdots & -\frac{1}{m} \\ -\frac{1}{m} & 1 - \frac{1}{m} & \cdots & -\frac{1}{m} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{m} & -\frac{1}{m} & \cdots & 1 - \frac{1}{m} \end{bmatrix}$$

Functional model - Example



Conclusion

- Graphical lasso is a method for structure learning that allows simplifying modeling distributions.
  - Particularly good in cases where $p > n$.
  - Otherwise, alternatives like the PC-Algorithm are sufficient.
- The mixed graphical model generalizes it to mixed data, with the limitation of only pairwise interactions.
- Because the likelihood involves a NP-hard problem, the optimization is performed using pseudo-likelihood.
- Extensions and improvements for the presented methods were done in the past years.