Faculty of Science

Structure estimation for Gaussian graphical models

Steffen Lauritzen, University of Copenhagen
Department of Mathematical Sciences

Overview of lectures

Lecture 1 Markov Properties and the Multivariate Gaussian Distribution

Lecture 2 Likelihood Analysis of Gaussian Graphical Models

Lecture 3 *Structure Estimation for Gaussian Graphical Models.*

For reference, if nothing else is mentioned, see Lauritzen (1996), Chapters 3 and 4.

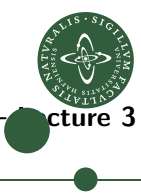
Gaussian graphical model

$\mathcal{S}(\mathcal{G})$ denotes the symmetric matrices A with $a_{\alpha\beta} = 0$ unless $\alpha \sim \beta$ and $\mathcal{S}^+(\mathcal{G})$ their positive definite elements.

Gaussian graphical model

$\mathcal{S}(\mathcal{G})$ denotes the symmetric matrices A with $a_{\alpha\beta} = 0$ unless $\alpha \sim \beta$ and $\mathcal{S}^+(\mathcal{G})$ their positive definite elements.

A *Gaussian graphical model* for X specifies X as multivariate normal with $K \in \mathcal{S}^+(\mathcal{G})$ and otherwise unknown.



Gaussian graphical model

$\mathcal{S}(\mathcal{G})$ denotes the symmetric matrices A with $a_{\alpha\beta} = 0$ unless $\alpha \sim \beta$ and $\mathcal{S}^+(\mathcal{G})$ their positive definite elements.

A *Gaussian graphical model* for X specifies X as multivariate normal with $K \in \mathcal{S}^+(\mathcal{G})$ and otherwise unknown.

The likelihood function based on a sample of size n is

$$L(K) \propto (\det K)^{n/2} e^{-\operatorname{tr}(Kw)/2},$$

where w is the (Wishart) matrix of sums of squares and products and $\Sigma^{-1} = K \in \mathcal{S}^+(\mathcal{G})$.

Representation via basis matrices

Define the matrices $T^u, u \in V \cup E$ as those with elements

$$T^u_{ij} = \begin{cases} 1 & \text{if } u \in V \text{ and } i = j = u \\ 1 & \text{if } u \in E \text{ and } u = \{i, j\} \\ 0 & \text{otherwise.} \end{cases};$$

then $T^u, u \in V \cup E$ *forms a basis* for the linear space $\mathcal{S}(\mathcal{G})$ of symmetric matrices over V which have zero entries ij whenever i and j are non-adjacent in $\mathcal{G}$.

Representation via basis matrices

Define the matrices $T^u, u \in V \cup E$ as those with elements

$$T^u_{ij} = \begin{cases} 1 & \text{if } u \in V \text{ and } i = j = u \\ 1 & \text{if } u \in E \text{ and } u = \{i, j\} \\ 0 & \text{otherwise.} \end{cases};$$

then $T^u, u \in V \cup E$ *forms a basis* for the linear space $\mathcal{S}(\mathcal{G})$ of symmetric matrices over V which have zero entries ij whenever i and j are non-adjacent in $\mathcal{G}$.

We can then identify the family as a (regular and canonical) exponential family with $-\operatorname{tr}(T^u W)/2, u \in V \cup E$ as canonical sufficient statistics.

This yields the *likelihood equations*

$$\operatorname{tr}(T^u w) = n \operatorname{tr}(T^u \Sigma), \quad u \in V \cup E.$$

Iterative Proportional Scaling

For $K \in \mathcal{S}^+(\mathcal{G})$ and $c \in \mathcal{C}$, define the operation of *adjusting the c-marginal* as follows: Let $a = V \setminus c$ and

$$M_c K = \begin{pmatrix} n(w_{cc})^{-1} + K_{ca}(K_{aa})^{-1}K_{ac} & K_{ca} \\ K_{ac} & K_{aa} \end{pmatrix}. \quad (1)$$

This operation is clearly well defined if w_{cc} is positive definite.

Iterative Proportional Scaling

For $K \in \mathcal{S}^+(\mathcal{G})$ and $c \in \mathcal{C}$, define the operation of *adjusting the c-marginal* as follows: Let $a = V \setminus c$ and

$$M_c K = \begin{pmatrix} n(w_{cc})^{-1} + K_{ca}(K_{aa})^{-1}K_{ac} & K_{ca} \\ K_{ac} & K_{aa} \end{pmatrix}. \quad (1)$$

This operation is clearly well defined if w_{cc} is positive definite.

Next we choose any ordering $(c_1, \ldots, c_k)$ of the cliques in $\mathcal{G}$. Choose further $K_0 = I$ and define for $r = 0, 1, \ldots$

$$K_{r+1} = (M_{c_1} \cdots M_{c_k})K_r.$$

Iterative Proportional Scaling

For $K \in \mathcal{S}^+(\mathcal{G})$ and $c \in \mathcal{C}$, define the operation of *adjusting the c-marginal* as follows: Let $a = V \setminus c$ and

$$M_c K = \begin{pmatrix} n(w_{cc})^{-1} + K_{ca}(K_{aa})^{-1}K_{ac} & K_{ca} \\ K_{ac} & K_{aa} \end{pmatrix}. \quad (1)$$

This operation is clearly well defined if w_{cc} is positive definite.

Next we choose any ordering $(c_1, \ldots, c_k)$ of the cliques in $\mathcal{G}$. Choose further $K_0 = I$ and define for $r = 0, 1, \ldots$

$$K_{r+1} = (M_{c_1} \cdots M_{c_k})K_r.$$

Then we have:

$$\hat{K} = \lim_{r \to \infty} K_r,$$

provided the maximum likelihood estimate $\hat{K}$ of K exists.

Characterizing decomposable graphs

A graph is *chordal* if all cycles of length ≥ 4 have chords.

The following are equivalent for any undirected graph $\mathcal{G}$.

(i) *$\mathcal{G}$ is chordal;*

(ii) *$\mathcal{G}$ is decomposable;*

(iii) *All maximal prime subgraphs of $\mathcal{G}$ are cliques;*

There are also many other useful characterizations of chordal graphs and algorithms that identify them.

Trees are chordal graphs and thus decomposable.

If the graph $\mathcal{G}$ is chordal, we say that the graphical model is *decomposable*.

In this case, *the IPS-algorithm converges in a finite number of steps*.

We also have the *factorization of densities*

$$f(x \mid \Sigma) = \frac{\prod_{C \in \mathcal{C}} f(x_C \mid \Sigma_C)}{\prod_{S \in \mathcal{S}} f(x_S \mid \Sigma_S)^{\nu(S)}} \tag{2}$$

where $\nu(S)$ is the number of times S appear as intersection between neighbouring cliques of a junction tree for $\mathcal{C}$.

Similar factorizations naturally hold for the maximum likelihood estimate $\hat{\Sigma}$.

Structure estimation

Advances in computing has set focus on *estimation of structure*:

- Model selection (e.g. subset selection in regression)

Structure estimation

Advances in computing has set focus on *estimation of structure*:

- Model selection (e.g. subset selection in regression)
- System identification (engineering)

Structure estimation

Advances in computing has set focus on *estimation of structure*:

- Model selection (e.g. subset selection in regression)
- System identification (engineering)
- Structural learning (AI or machine learning)

Structure estimation

Advances in computing has set focus on *estimation of structure*:

- Model selection (e.g. subset selection in regression)
- System identification (engineering)
- Structural learning (AI or machine learning)

Graphical models describe conditional independence structures, so good case for formal analysis.

Structure estimation

Advances in computing has set focus on *estimation of structure*:

- Model selection (e.g. subset selection in regression)
- System identification (engineering)
- Structural learning (AI or machine learning)

Graphical models describe conditional independence structures, so good case for formal analysis.

Methods must scale well with data size, as *many* structures and *huge* collections of data are to be considered.

Why estimation of structure?

- Parallel to e.g. density estimation

Why estimation of structure?

- Parallel to e.g. density estimation
- Obtain quick overview of relations between variables in complex systems

Why estimation of structure?

- Parallel to e.g. density estimation
- Obtain quick overview of relations between variables in complex systems
- Data mining

Why estimation of structure?

- Parallel to e.g. density estimation
- Obtain quick overview of relations between variables in complex systems
- Data mining
- Gene regulatory networks

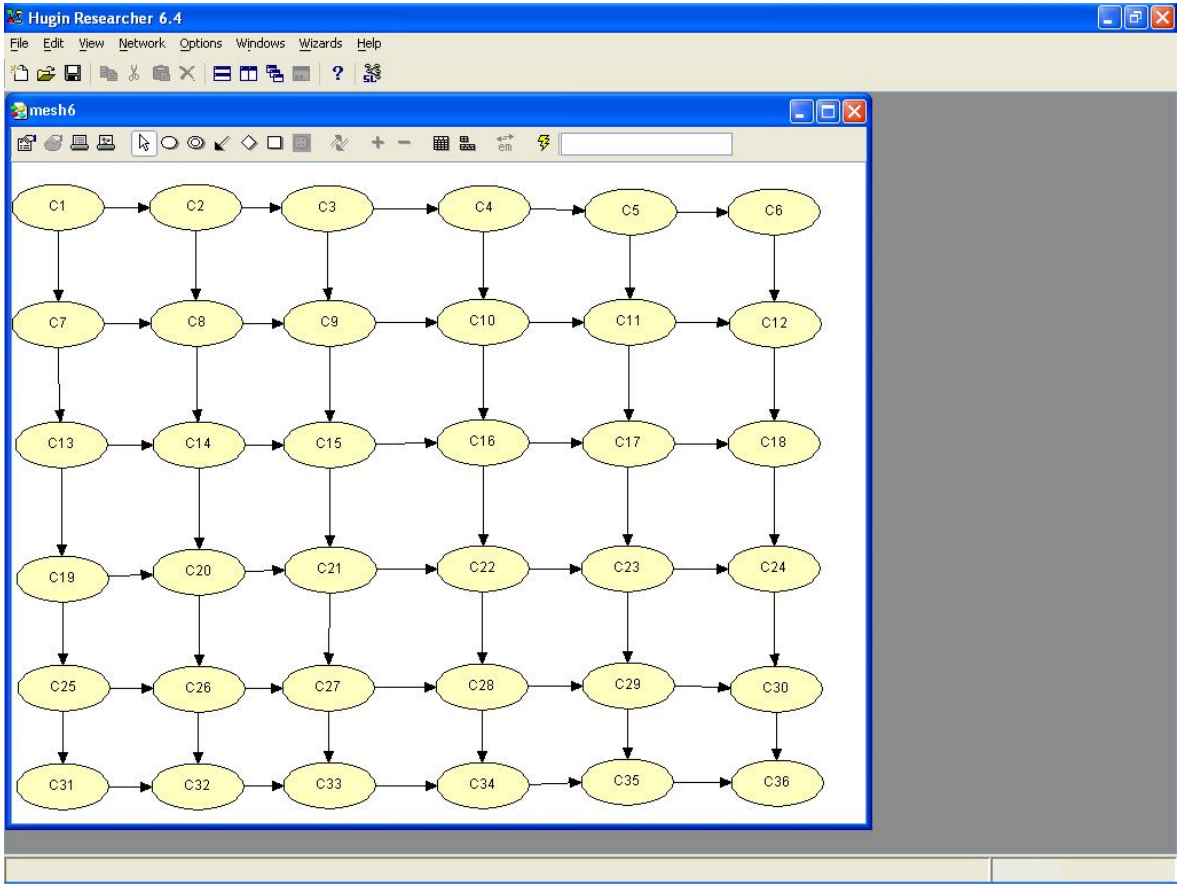
Why estimation of structure?

- Parallel to e.g. density estimation
- Obtain quick overview of relations between variables in complex systems
- Data mining
- Gene regulatory networks
- Reconstructing family trees from DNA information

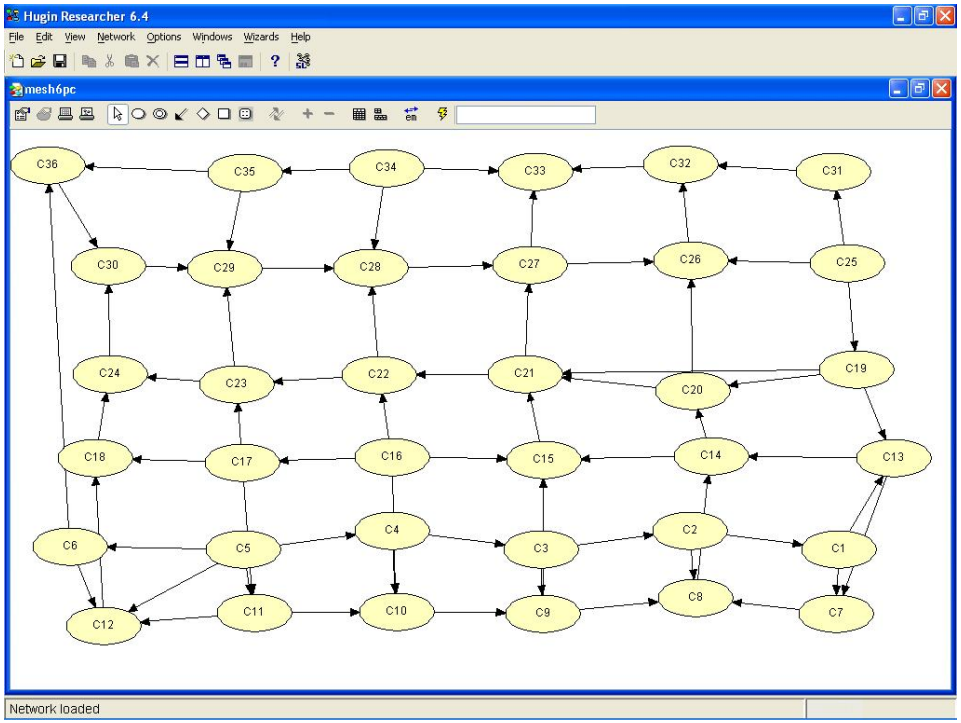
Why estimation of structure?

- Parallel to e.g. density estimation
- Obtain quick overview of relations between variables in complex systems
- Data mining
- Gene regulatory networks
- Reconstructing family trees from DNA information
- General interest in sparsity.

Markov mesh model

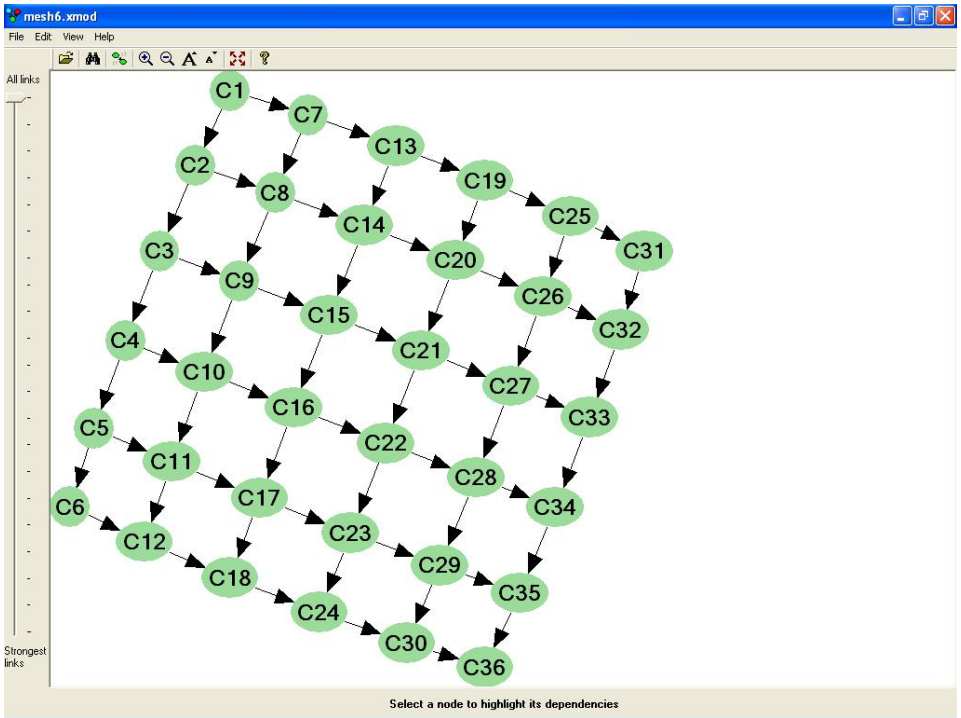


PC algorithm



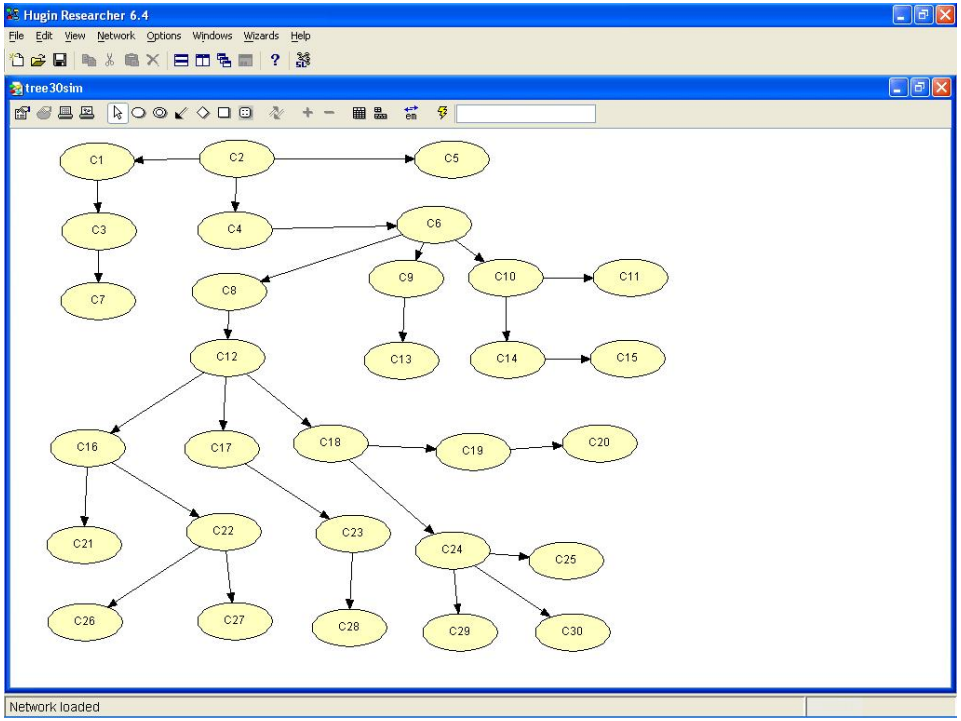
Crudest algorithm (HUGIN), 10000 simulated cases

Bayesian GES



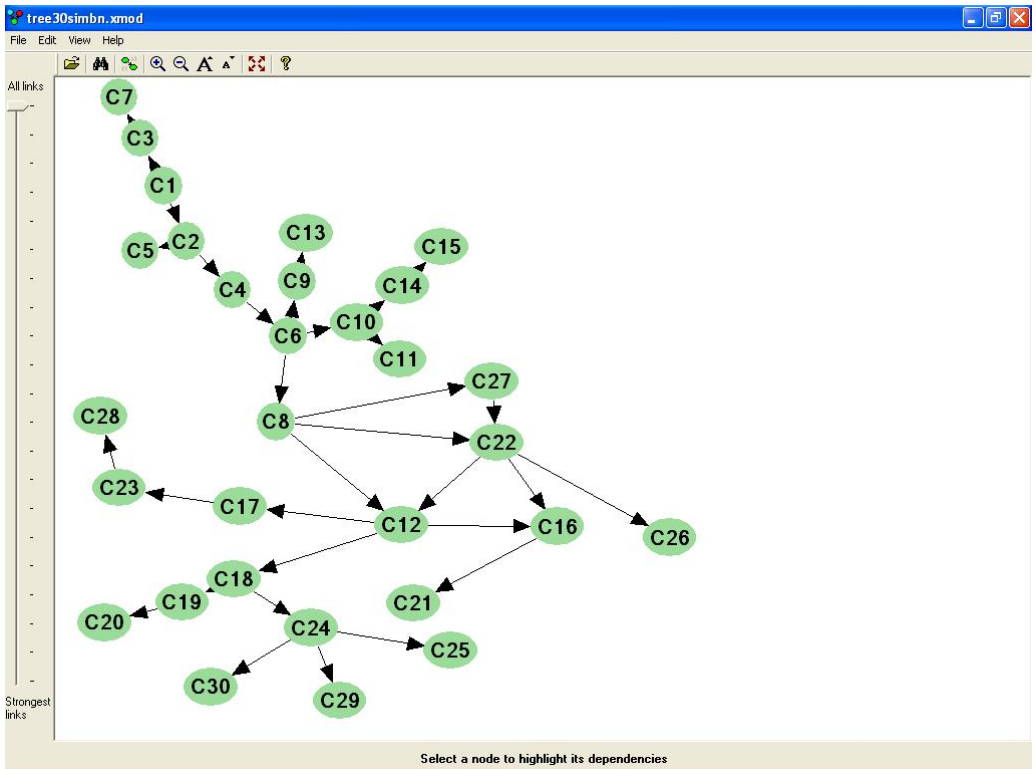
Crudest algorithm (WinMine), 10000 simulated cases

Tree model

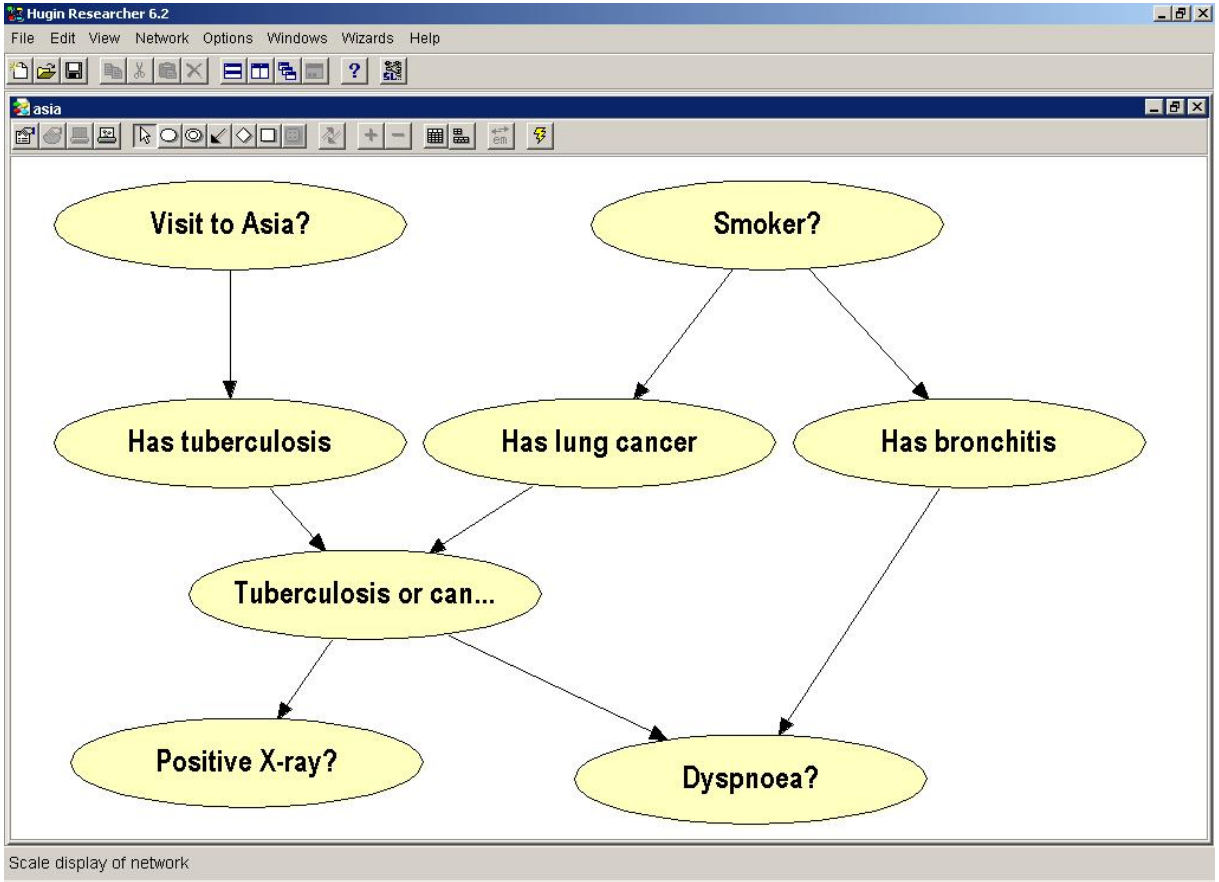


PC algorithm, 10000 cases, correct reconstruction

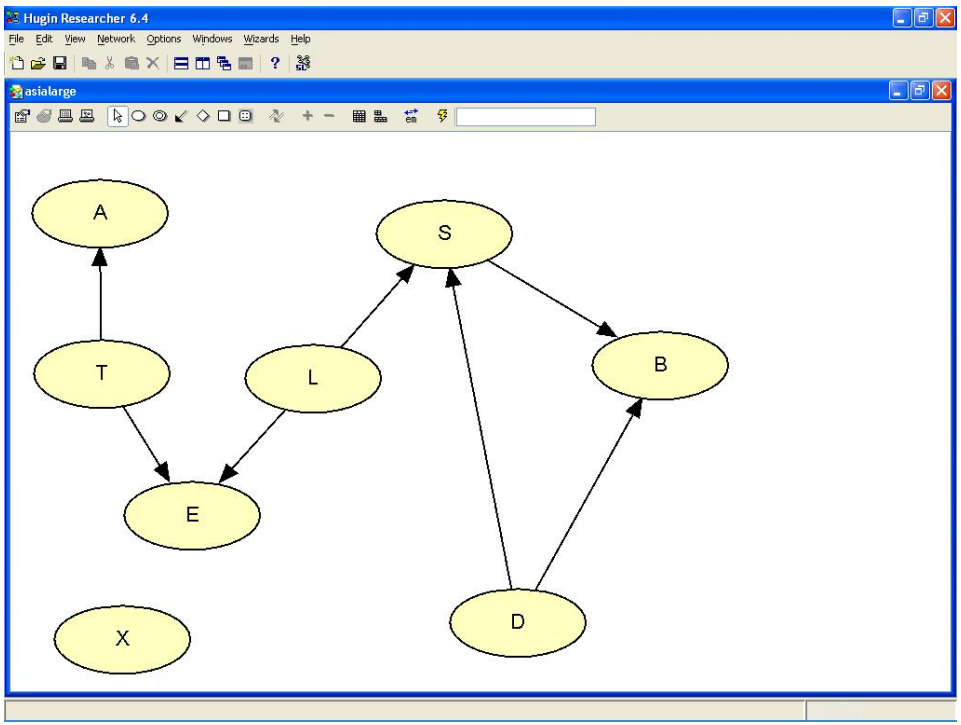
Bayesian GES on tree



Chest clinic

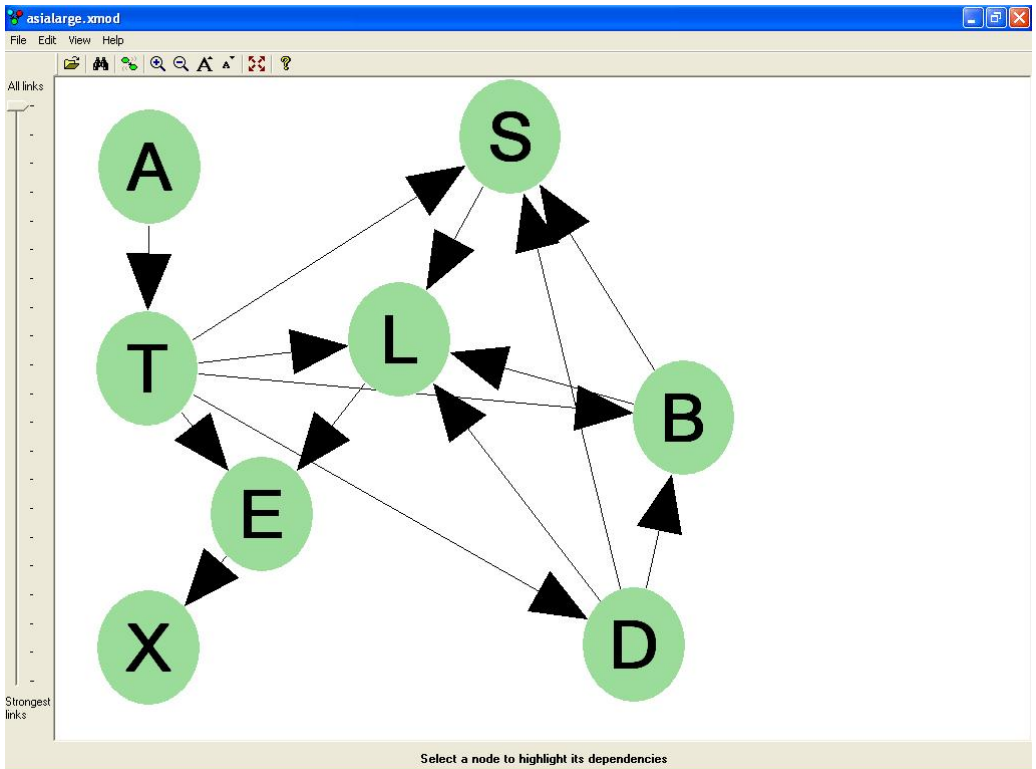


PC algorithm

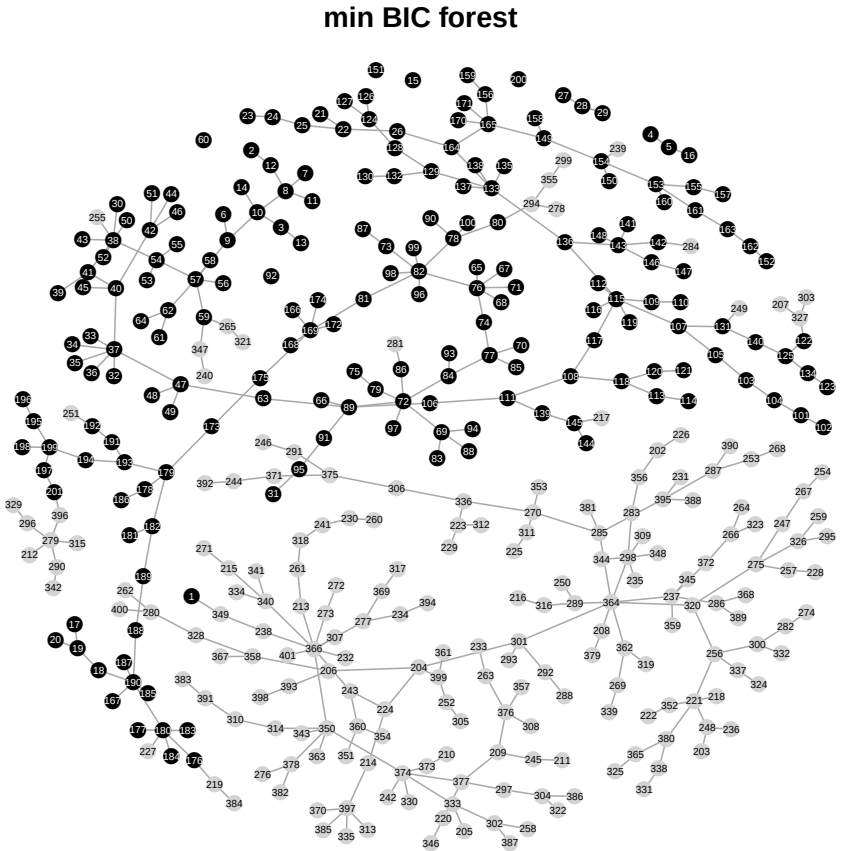


10000 simulated cases

Bayesian GES



SNPs and gene expressions



Methods for structure identification in graphical models can be classified into three types:

- *score-based methods:* For example optimizing a penalized likelihood by using convex programming e.g. `glasso`;

Methods for structure identification in graphical models can be classified into three types:

- *score-based methods:* For example optimizing a penalized likelihood by using convex programming e.g. `glasso`;
- *Bayesian methods:* Identifying posterior distributions over graphs; can also use posterior probability as score.

Methods for structure identification in graphical models can be classified into three types:

- *score-based methods:* For example optimizing a penalized likelihood by using convex programming e.g. `glasso`;
- *Bayesian methods:* Identifying posterior distributions over graphs; can also use posterior probability as score.
- constraint-based methods: Querying conditional independences and identifying compatible independence structures, for example PC, PC*, NPC, IC, CI, FCI, SIN, QP, ...

Penalized likelihood

Methods based on pure maximum likelihood are not feasible when the dimension of the parameter space varies.

Penalized likelihood

Methods based on pure maximum likelihood are not feasible when the dimension of the parameter space varies.

Trade off goodness-of-fit, measured by the maximized likelihood, against the complexity of the model.

Penalized likelihood

Methods based on pure maximum likelihood are not feasible when the dimension of the parameter space varies.

Trade off goodness-of-fit, measured by the maximized likelihood, against the complexity of the model.

$$IC_\kappa(\mathcal{G}) = -2 \log L_{\mathcal{G}}(\hat{\theta}_{\mathcal{G}}) + \kappa \dim(\mathcal{G}),$$

$\hat{\theta}_{\mathcal{G}}$ is the MLE, $\dim(\mathcal{G})$ is the number of free parameters, and κ is a constant that gives the exchange rate for trading fit and parameters.

Penalized likelihood

Methods based on pure maximum likelihood are not feasible when the dimension of the parameter space varies.

Trade off goodness-of-fit, measured by the maximized likelihood, against the complexity of the model.

$$IC_\kappa(\mathcal{G}) = -2 \log L_{\mathcal{G}}(\hat{\theta}_{\mathcal{G}}) + \kappa \dim(\mathcal{G}),$$

$\hat{\theta}_{\mathcal{G}}$ is the MLE, $\dim(\mathcal{G})$ is the number of free parameters, and κ is a constant that gives the exchange rate for trading fit and parameters.

κ may depend on the number n of observations, but is constant over the set of possible graphs $\mathfrak{G}$.

Akaike's Information Criterion

The criterion AIC has $\kappa = 2$ independently of the number of observations. It is meant to optimize the prediction error for predicting the next observation.

Akaike's Information Criterion

The criterion AIC has $\kappa = 2$ independently of the number of observations. It is meant to optimize the prediction error for predicting the next observation.

AIC is not consistent for $n \to \infty$ as it will tend to have too many parameters.

Akaike's Information Criterion

The criterion AIC has $\kappa = 2$ independently of the number of observations. It is meant to optimize the prediction error for predicting the next observation.

AIC is not consistent for $n \to \infty$ as it will tend to have too many parameters.

Hence used for a Gaussian graphical model for large n, the model will tend not to be sparse.

Bayesian Information Criterion

An asymptotic Bayesian argument leads to BIC, which has $\kappa = \log n$, where n is the number of observations.

Bayesian Information Criterion

An asymptotic Bayesian argument leads to BIC, which has $\kappa = \log n$, where n is the number of observations.

The BIC ensures consistent estimation of the graph.

Bayesian Information Criterion

An asymptotic Bayesian argument leads to BIC, which has $\kappa = \log n$, where n is the number of observations.

The BIC ensures consistent estimation of the graph.

However, the true structure can be identified faster if, say

$$\kappa_n = C \log \log n$$

for some $C > 1$.

Other penalized methods

Other penalized likelihood methods use criteria of the form

$$\ell_\kappa(\mathcal{G}, \theta_\mathcal{G}) = -2 \log L_\mathcal{G}(\theta_\mathcal{G}) + \kappa ||\theta_\mathcal{G}||,$$

where $||\theta_\mathcal{G}||$ is measuring the size of the parameter, for example using a vector space norm.

Other penalized methods

Other penalized likelihood methods use criteria of the form

$$\ell_\kappa(\mathcal{G}, \theta_\mathcal{G}) = -2 \log L_\mathcal{G}(\theta_\mathcal{G}) + \kappa ||\theta_\mathcal{G}||,$$

where $||\theta_\mathcal{G}||$ is measuring the size of the parameter, for example using a vector space norm.

An example of this for Gaussian graphical models is the so-called *graphical lasso* based on minimizing

$$\ell_\kappa(K) = -2 \log L(K) + \kappa ||K||_1$$

where now the graph $\mathcal{G}$ is only implicitly represented through K itself.

Other penalized methods

Other penalized likelihood methods use criteria of the form

$$\ell_\kappa(\mathcal{G}, \theta_\mathcal{G}) = -2 \log L_\mathcal{G}(\theta_\mathcal{G}) + \kappa ||\theta_\mathcal{G}||,$$

where $||\theta_\mathcal{G}||$ is measuring the size of the parameter, for example using a vector space norm.

An example of this for Gaussian graphical models is the so-called *graphical lasso* based on minimizing

$$\ell_\kappa(K) = -2 \log L(K) + \kappa ||K||_1$$

where now the graph $\mathcal{G}$ is only implicitly represented through K itself.

This is a convex optimization problem and in some sense ℓ_κ is a convex variant of the IC criteria.

Bayesian methods

A full Bayesian approach will use suitable prior distributions, in the Gaussian case known as *hyper Markov Wishart* and *hyper Markov inverse Wishart* prior distributions.

Bayesian methods

A full Bayesian approach will use suitable prior distributions, in the Gaussian case known as *hyper Markov Wishart* and *hyper Markov inverse Wishart* prior distributions.

One then writes:

$$f(x \mid \mathcal{G}) = \int_{K \in \mathcal{S}(\mathcal{G})^+} f(x \mid K) \pi_{\mathcal{G}}(dK)$$

Bayesian methods

A full Bayesian approach will use suitable prior distributions, in the Gaussian case known as *hyper Markov Wishart* and *hyper Markov inverse Wishart* prior distributions.

One then writes:

$$f(x \mid \mathcal{G}) = \int_{K \in \mathcal{S}(\mathcal{G})^+} f(x \mid K) \pi_{\mathcal{G}}(dK)$$

and further

$$\pi(\mathcal{G} \mid x) \propto f(x \mid \mathcal{G}) \pi(\mathcal{G}).$$

Bayesian methods

A full Bayesian approach will use suitable prior distributions, in the Gaussian case known as *hyper Markov Wishart* and *hyper Markov inverse Wishart* prior distributions.

One then writes:

$$f(x \mid \mathcal{G}) = \int_{K \in \mathcal{S}(\mathcal{G})^+} f(x \mid K) \pi_{\mathcal{G}}(dK)$$

and further

$$\pi(\mathcal{G} \mid x) \propto f(x \mid \mathcal{G}) \pi(\mathcal{G}).$$

Attempting, say, to maximize $\pi(\mathcal{G} \mid x)$ over $\mathcal{G}$ leads to the *MAP* estimate of $\mathcal{G}$.

Bayesian methods

A full Bayesian approach will use suitable prior distributions, in the Gaussian case known as *hyper Markov Wishart* and *hyper Markov inverse Wishart* prior distributions.

One then writes:

$$f(x \mid \mathcal{G}) = \int_{K \in \mathcal{S}(\mathcal{G})^+} f(x \mid K) \pi_{\mathcal{G}}(dK)$$

and further

$$\pi(\mathcal{G} \mid x) \propto f(x \mid \mathcal{G}) \pi(\mathcal{G}).$$

Attempting, say, to maximize $\pi(\mathcal{G} \mid x)$ over $\mathcal{G}$ leads to the *MAP* estimate of $\mathcal{G}$.

Asymptotically for large n this would be equivalent to BIC.

Estimating trees and forests

The simplest case to consider is the case where the unknown conditional independence structure is a tree $\mathcal{T} \in \mathfrak{T}(V)$;

Estimating trees and forests

The simplest case to consider is the case where the unknown conditional independence structure is a tree $\mathcal{T} \in \mathfrak{T}(V)$;

since a tree is decomposable, any distribution P which factorizes w.r.t. $\mathcal{T} = (V, E)$ has a density of the form

$$f(x) = \frac{\prod_{e \in E} f_e(x_e)}{\prod_{v \in V} f_v(x_v)^{d(v)-1}} = \prod_{uv \in E} \frac{f_{uv}(x_{uv})}{f_u(x_u) f_v(x_v)} \prod_{v \in V} f_v(x_v). \tag{3}$$

Maximum likelihood trees

Next we shall consider the situation where we have a sample $x = (x^1, \ldots, x^n)$ from a distribution P of $X = X_V$ which is Gaussian and is known to factorize according to a tree $\mathcal{T} \in \mathfrak{T}(V)$ but both P and $\mathcal{T}$ is otherwise unknown.

Maximum likelihood trees

Next we shall consider the situation where we have a sample $x = (x^1, \ldots, x^n)$ from a distribution P of $X = X_V$ which is Gaussian and is known to factorize according to a tree $\mathcal{T} \in \mathfrak{T}(V)$ but both P and $\mathcal{T}$ is otherwise unknown.

In other words, we assume the unknown concentration matrix K satisfies

$$K \in \cup_{\mathcal{T} \in \mathfrak{T}(V)} \mathcal{S}^+(\mathcal{T}).$$

Maximum likelihood trees

Next we shall consider the situation where we have a sample $x = (x^1, \ldots, x^n)$ from a distribution P of $X = X_V$ which is Gaussian and is known to factorize according to a tree $\mathcal{T} \in \mathfrak{T}(V)$ but both P and $\mathcal{T}$ is otherwise unknown.

In other words, we assume the unknown concentration matrix K satisfies

$$K \in \cup_{\mathcal{T} \in \mathfrak{T}(V)} \mathcal{S}^+(\mathcal{T}).$$

To maximize the likelihood function over this parameter space, we first maximize for a fixed tree to get the *profile likelihood* $\hat{L}(\mathcal{T} \,|\, x)$, where

$$\hat{L}(\mathcal{T}) = \hat{L}(\mathcal{T} \,|\, x) = \sup_{K \in \mathcal{S}^+(\mathcal{T})} L(K \,|\, x);$$

Maximum likelihood trees

Next we shall consider the situation where we have a sample $x = (x^1, \ldots, x^n)$ from a distribution P of $X = X_V$ which is Gaussian and is known to factorize according to a tree $\mathcal{T} \in \mathfrak{T}(V)$ but both P and $\mathcal{T}$ is otherwise unknown.

In other words, we assume the unknown concentration matrix K satisfies

$$K \in \cup_{\mathcal{T} \in \mathfrak{T}(V)} \mathcal{S}^+(\mathcal{T}).$$

To maximize the likelihood function over this parameter space, we first maximize for a fixed tree to get the *profile likelihood* $\hat{L}(\mathcal{T} \,|\, x)$, where

$$\hat{L}(\mathcal{T}) = \hat{L}(\mathcal{T} \,|\, x) = \sup_{K \in \mathcal{S}^+(\mathcal{T})} L(K \,|\, x);$$

we then further maximize $\hat{L}(\mathcal{T})$ over all trees $\mathcal{T} \in \mathfrak{T}(V)$.

Since a tree is decomposable, the profile likelihood satisfies

$$\begin{aligned}\hat{L}(\mathcal{T} \,|\, x) &= f(x \,|\, \hat{K}_{\mathcal{T}}) \\ &= \frac{\prod_{e\in E} \hat{f}_{[e]}(x_e)}{\prod_{v\in V} \hat{f}_{[v]}(x_v)^{d(v)-1}} \\ &= \prod_{uv\in E} \frac{\hat{f}_{[uv]}(x_{uv})}{\hat{f}_{[u]}(x_u)\hat{f}_{[v]}(x_v)} \prod_{v\in V} \hat{f}_{[v]}(x_v).\end{aligned}$$

Since a tree is decomposable, the profile likelihood satisfies

$$\begin{aligned}\hat{L}(\mathcal{T} \,|\, x) &= f(x \,|\, \hat{K}_{\mathcal{T}}) \\ &= \frac{\prod_{e \in E} \hat{f}_{[e]}(x_e)}{\prod_{v \in V} \hat{f}_{[v]}(x_v)^{d(v)-1}} \\ &= \prod_{uv \in E} \frac{\hat{f}_{[uv]}(x_{uv})}{\hat{f}_{[u]}(x_u)\hat{f}_{[v]}(x_v)} \prod_{v \in V} \hat{f}_{[v]}(x_v).\end{aligned}$$

Here $\hat{f}_{[A]}(x)$ denotes the maximized likelihood for the marginal distribution of X_A based on data x_a only and using the saturated model for X_A.

More precisely, for $x = (x^1, \ldots, x^n)$ we have

$$\begin{aligned}\hat{f}_{[uv]}(x) &= (2\pi)^{-n} \det(W_{\{uv\}}/n)^{-n/2} e^{-\operatorname{tr}\left\{n(W_{\{uv\}})^{-1} W_{\{uv\}}\right\}/2} \\ &= n^n (2\pi)^{-n} (w_{uu} w_{vv} - w_{uv}^2)^{-n/2} \exp(-n)\end{aligned}$$

and

$$\begin{aligned}\hat{f}_{[v]}(x) &= (2\pi)^{-n/2} (W_{vv}/n)^{-n/2} \exp(-n/2) \\ &= n^{n/2} (2\pi)^{-n/2} (w_{vv})^{-n/2} \exp(-n/2),\end{aligned}$$

where $W = \{w_{uv}, u, v \in V\}$ is the Wishart matrix of sums and squares of products.

Thus we get in particular

$$\log \frac{\hat{f}_{[uv]}(x_{uv})}{\hat{f}_{[u]}(x_u)\hat{f}_{[v]}(x_v)} = -\frac{n}{2}\log\frac{w_{uu}w_{vv}-w_{uv}^2}{w_{uu}w_{vv}} = -\frac{n}{2}\log(1-r_{uv}^2)$$

where r_{uv} is the *empirical correlation coefficient*

$$r_{uv} = w_{uv}/\sqrt{w_{uu}w_{vv}}.$$

Define the *empirical correlation weight* ω_{uv} of the edge uv as

$$\omega_{uv} = -\frac{n}{2}\log(1 - r_{uv}^2)$$

and let

$$\omega(\mathcal{T}) = \sum_{uv \in E(\mathcal{T})} \omega_{uv}$$

denote the total empirical weight of the tree $\mathcal{T}$.

Define the *empirical correlation weight* ω_{uv} of the edge uv as

$$\omega_{uv} = -\frac{n}{2}\log(1 - r_{uv}^2)$$

and let

$$\omega(\mathcal{T}) = \sum_{uv \in E(\mathcal{T})} \omega_{uv}$$

denote the total empirical weight of the tree $\mathcal{T}$.

The matrix $\Omega = \{\omega_{uv}\}$ is the *correlation weight matrix.*

Further let $\hat{L}(\emptyset)$ denote the maximized likelihood under independence

$$\hat{L}(\emptyset) = \prod_{v \in V} \hat{f}_{[v]}(x_v).$$

Further let $\hat{L}(\emptyset)$ denote the maximized likelihood under independence

$$\hat{L}(\emptyset) = \prod_{v \in V} \hat{f}_{[v]}(x_v).$$

Then, clearly, it holds that

$$\log \hat{L}(\mathcal{T}) - \log \hat{L}(\emptyset) = \omega(\mathcal{T}) = \sum_{uv \in E(\mathcal{T})} \omega_{uv}. \qquad (4)$$

We say that $\hat{\mathcal{T}}$ is a *maximum likelihood tree* based on a sample $x = x^1, \ldots, x^n$ if $\hat{\mathcal{T}}$ satisfies

$$L(\hat{\mathcal{T}}) = \sup_{\mathcal{T} \in \mathfrak{T}(V)} \hat{L}(\mathcal{T}).$$

We say that $\hat{\mathcal{T}}$ is a *maximum likelihood tree* based on a sample $x = x^1, \ldots, x^n$ if $\hat{\mathcal{T}}$ satisfies

$$L(\hat{\mathcal{T}}) = \sup_{\mathcal{T} \in \mathfrak{T}(V)} \hat{L}(\mathcal{T}).$$

A *spanning tree* $\mathcal{T}$ of a connected $\mathcal{G}(V, E)$ is a subtree $\mathcal{T} = (V, E_{\mathcal{T}})$ of $\mathcal{G}$ which has the same vertex set and is a tree. That is, $E_{\mathcal{T}} \subseteq E(\mathcal{G})$.

We then have the following result:

Theorem

A tree $\hat{\mathcal{T}}_* \in \mathfrak{T}(V)$ is a maximum likelihood tree if and only if it is a maximum weight spanning tree (MWST) of the complete graph with vertex set V for the weight matrix Ω with

$$\omega_{uv} = -\frac{n}{2}\log(1 - r_{uv}^2)$$

that is

$$\hat{\mathcal{T}} = \underset{\mathcal{T}\in\mathfrak{T}(V)}{\arg\max}\, \hat{L}(\mathcal{T}) \iff \hat{\mathcal{T}} = \underset{\mathcal{T}\in\mathfrak{T}(V)}{\arg\max}\, \omega(\mathcal{T}).$$

Kruskal's algorithm

This runs as follows and outputs a MWST:

Kruskal's algorithm

This runs as follows and outputs a MWST: smallskip
Order all off-diagonal elements in the matrix Ω from largest to smallest so that for $E = \{e_1, \ldots, e_k\}$ where $k = |V|(V-1)/2$ $\omega_{e_i} \geq \omega_{e_j}$ whenever $i > j$.

Kruskal's algorithm

This runs as follows and outputs a MWST: smallskip
Order all off-diagonal elements in the matrix Ω from largest to smallest so that for $E = \{e_1, \ldots, e_k\}$ where $k = |V|(V-1)/2$ $\omega_{e_i} \geq \omega_{e_j}$ whenever $i > j$.

1. Let $\mathcal{F} = (V, \emptyset)$

Kruskal's algorithm

This runs as follows and outputs a MWST: smallskip
Order all off-diagonal elements in the matrix Ω from largest to smallest so that for $E = \{e_1, \ldots, e_k\}$ where $k = |V|(V-1)/2$ $\omega_{e_i} \geq \omega_{e_j}$ whenever $i > j$.

1. Let $\mathcal{F} = (V, \emptyset)$
2. **for** $i = 1, i+1$ **until** $\mathcal{F}$ is a spanning tree **do**:
3. **if** $E(\mathcal{F}) \cup e_i$ is a forest let $E(\mathcal{F}) = E(\mathcal{F}) \cup e_i$, **else** let $E(\mathcal{F}) = E(\mathcal{F})$.

Kruskal's algorithm

This runs as follows and outputs a MWST: smallskip
Order all off-diagonal elements in the matrix Ω from largest to smallest so that for $E = \{e_1, \ldots, e_k\}$ where $k = |V|(V-1)/2$ $\omega_{e_i} \geq \omega_{e_j}$ whenever $i > j$.

1. Let $\mathcal{F} = (V, \emptyset)$
2. **for** $i = 1, i+1$ **until** $\mathcal{F}$ is a spanning tree **do**:
3. **if** $E(\mathcal{F}) \cup e_i$ is a forest let $E(\mathcal{F}) = E(\mathcal{F}) \cup e_i$, **else** let $E(\mathcal{F}) = E(\mathcal{F})$.
4. **return** $\mathcal{F}$.

Penalized likelihood forests

If we instead wish to estimate an unknown *forest*, i.e. assume that $K \in \mathcal{S}^+(\mathcal{F})$ where $\mathcal{F}$ is unknown, we use a penalized form of the likelihood:

$$IC_\kappa(\mathcal{F}) = -2 \log \hat{L}(\mathcal{F}) + \kappa\{|V| + |E(\mathcal{F})|\}$$

since $|V| + |E(\mathcal{F})|$ is the dimension of the model determined by $\mathcal{F}$.

Penalized likelihood forests

If we instead wish to estimate an unknown *forest*, i.e. assume that $K \in \mathcal{S}^+(\mathcal{F})$ where $\mathcal{F}$ is unknown, we use a penalized form of the likelihood:

$$IC_\kappa(\mathcal{F}) = -2 \log \hat{L}(\mathcal{F}) + \kappa\{|V| + |E(\mathcal{F})|\}$$

since $|V| + |E(\mathcal{F})|$ is the dimension of the model determined by $\mathcal{F}$.

Using (4) yields

$$\begin{aligned} IC_\kappa(\mathcal{F}) &= -2\{\Omega(\mathcal{F}) - \kappa|E(\mathcal{F})|/2\} + \text{const} \\ &= -2\left\{\sum_{uv \in E(\mathcal{F})} (\omega_{uv} - \kappa/2)\right\} + \text{const.} \end{aligned}$$

Thus we can minimize the IC-score by using a modification of Kruskal's algorithm on Ω^κ, where $\omega^\kappa_{uv} = \omega_{uv} - \kappa/2$:

Thus we can minimize the IC-score by using a modification of Kruskal's algorithm on Ω^κ, where $\omega^\kappa_{uv} = \omega_{uv} - \kappa/2$:

Discard all negative off-diagonal elements in the matrix Ω^κ and order the remaining from largest to smallest.

Thus we can minimize the IC-score by using a modification of Kruskal's algorithm on Ω^κ, where $\omega^\kappa_{uv} = \omega_{uv} - \kappa/2$:

Discard all negative off-diagonal elements in the matrix Ω^κ and order the remaining from largest to smallest.

1. Let $\mathcal{F} = (V, \emptyset)$

Thus we can minimize the IC-score by using a modification of Kruskal's algorithm on Ω^κ, where $\omega^\kappa_{uv} = \omega_{uv} - \kappa/2$:

Discard all negative off-diagonal elements in the matrix Ω^κ and order the remaining from largest to smallest.

1. Let $\mathcal{F} = (V, \emptyset)$
2. **for** $i = 1, i + 1$ **until** $\mathcal{F}$ is a spanning tree **do**:
3. **if** $E(\mathcal{F}) \cup e_i$ is a forest let $E(\mathcal{F}) = E(\mathcal{F}) \cup e_i$, **else** let $E(\mathcal{F}) = E(\mathcal{F})$.

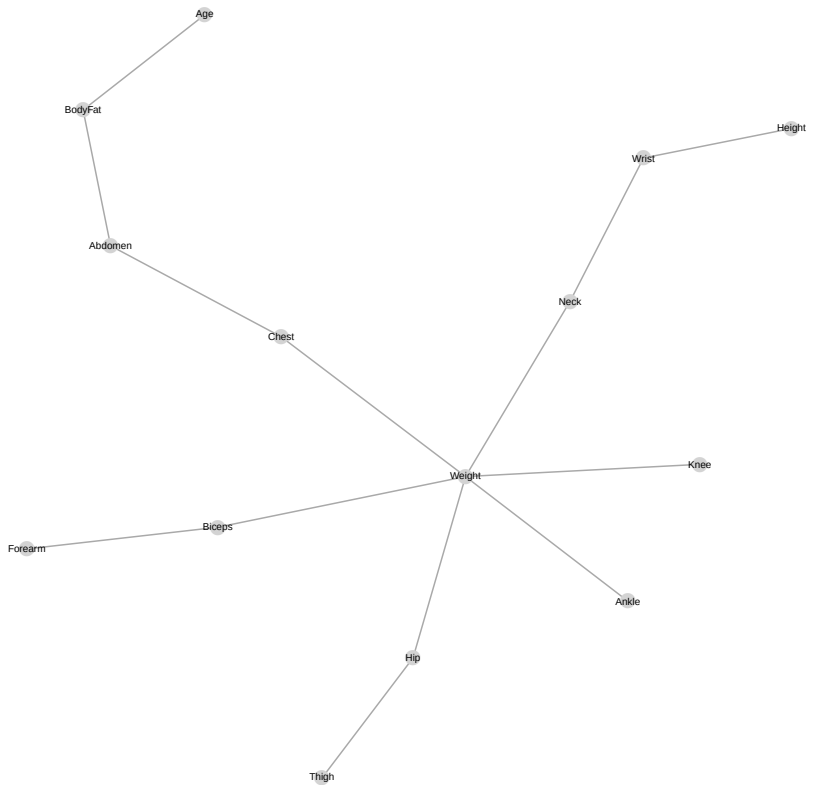
Thus we can minimize the IC-score by using a modification of Kruskal's algorithm on Ω^κ, where $\omega^\kappa_{uv} = \omega_{uv} - \kappa/2$:

Discard all negative off-diagonal elements in the matrix Ω^κ and order the remaining from largest to smallest.

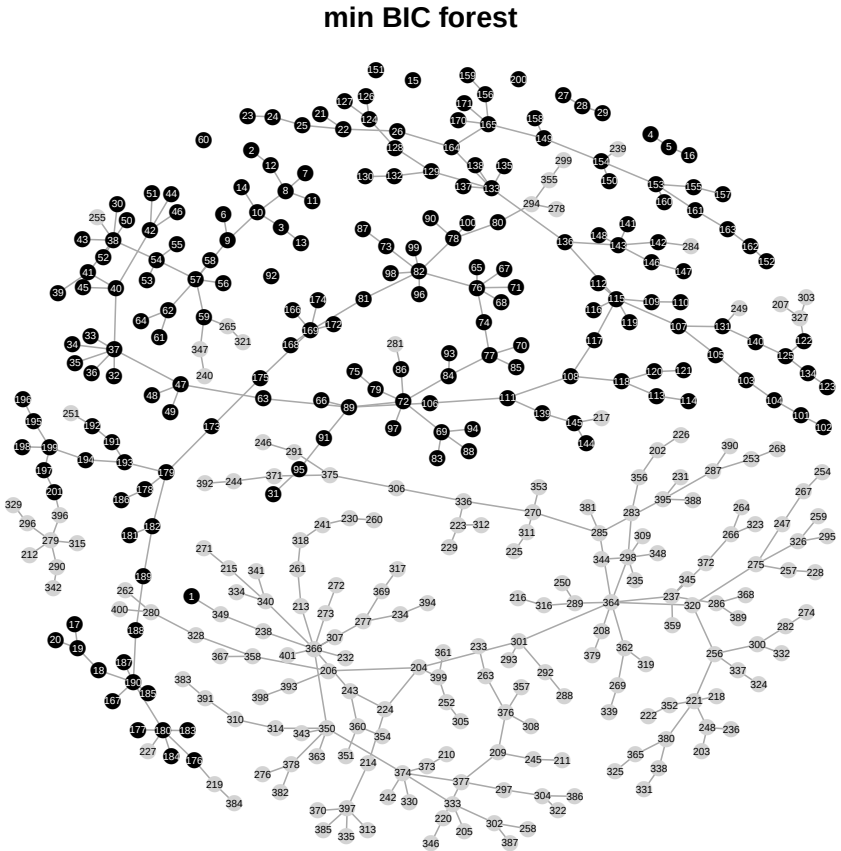
1. Let $\mathcal{F} = (V, \emptyset)$
2. **for** $i = 1, i + 1$ **until** $\mathcal{F}$ is a spanning tree **do**:
3. **if** $E(\mathcal{F}) \cup e_i$ is a forest let $E(\mathcal{F}) = E(\mathcal{F}) \cup e_i$, **else** let $E(\mathcal{F}) = E(\mathcal{F})$.
4. **return** $\mathcal{F}$.

BodyFat: min BIC forest

Data in Højsgaard et al. (2012). Measurements of body parts interesting for prediction of body fat.



SNPs and gene expressions



Random graphs for posterior analysis

A Bayesian approach to graphical model analysis implies setting up a prior distribution over a class of graphs, say undirected trees, and then finding the posterior distribution.

Random graphs for posterior analysis

A Bayesian approach to graphical model analysis implies setting up a prior distribution over a class of graphs, say undirected trees, and then finding the posterior distribution.

For example, if the prior is *uniform over trees,* and parameters are *hyper Markov* (Dawid and Lauritzen, 1993), the posterior distribution based on data x is

$$p^*(\tau \mid x) \propto w(\tau) = \prod_{e \in E(\tau)} \mathrm{BF}_e$$

where BF_e is the *Bayes factor for independence* among the variables at the endpoints of e;

Random graphs for posterior analysis

A Bayesian approach to graphical model analysis implies setting up a prior distribution over a class of graphs, say undirected trees, and then finding the posterior distribution.

For example, if the prior is *uniform over trees,* and parameters are *hyper Markov* (Dawid and Lauritzen, 1993), the posterior distribution based on data x is

$$p^*(\tau \mid x) \propto w(\tau) = \prod_{e \in E(\tau)} \mathrm{BF}_e$$

where BF_e is the *Bayes factor for independence* among the variables at the endpoints of e;

The unknown normalization constant $\sum_\tau w(\tau)$ can be found as a determinant using the *matrix tree theorem*.

For chordal graphs $\mathcal{G}$ the similar expression becomes

$$p^*(\mathcal{G} \mid x) \propto \frac{\prod_{C \in \mathcal{C}(\mathcal{G})} w(C \mid x)}{\prod_{S \in \mathcal{S}(\mathcal{G})} w(S \mid x)^{\nu_{\mathcal{G}}(S)}}, \tag{5}$$

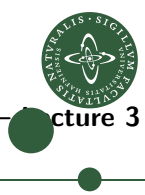
where $\mathcal{C}(\mathcal{G})$ are the maximal cliques of $\mathcal{G}$, $\mathcal{S}(\mathcal{G})$ the minimal complete separators, and $\nu_{\mathcal{G}}(S)$ are certain graph invariants.

For chordal graphs $\mathcal{G}$ the similar expression becomes

$$p^*(\mathcal{G} \mid x) \propto \frac{\prod_{C \in \mathcal{C}(\mathcal{G})} w(C \mid x)}{\prod_{S \in \mathcal{S}(\mathcal{G})} w(S \mid x)^{\nu_{\mathcal{G}}(S)}}, \tag{5}$$

where $\mathcal{C}(\mathcal{G})$ are the maximal cliques of $\mathcal{G}$, $\mathcal{S}(\mathcal{G})$ the minimal complete separators, and $\nu_{\mathcal{G}}(S)$ are certain graph invariants.

How can posterior distributions of this form be represented and/or simulated and what are the properties of such distributions?



For chordal graphs $\mathcal{G}$ the similar expression becomes

$$p^*(\mathcal{G} \mid x) \propto \frac{\prod_{C \in \mathcal{C}(\mathcal{G})} w(C \mid x)}{\prod_{S \in \mathcal{S}(\mathcal{G})} w(S \mid x)^{\nu_{\mathcal{G}}(S)}}, \tag{5}$$

where $\mathcal{C}(\mathcal{G})$ are the maximal cliques of $\mathcal{G}$, $\mathcal{S}(\mathcal{G})$ the minimal complete separators, and $\nu_{\mathcal{G}}(S)$ are certain graph invariants.

How can posterior distributions of this form be represented and/or simulated and what are the properties of such distributions?

Even case where the graphs considered is all forests is difficult

For chordal graphs $\mathcal{G}$ the similar expression becomes

$$p^*(\mathcal{G} \mid x) \propto \frac{\prod_{C \in \mathcal{C}(\mathcal{G})} w(C \mid x)}{\prod_{S \in \mathcal{S}(\mathcal{G})} w(S \mid x)^{\nu_{\mathcal{G}}(S)}}, \tag{5}$$

where $\mathcal{C}(\mathcal{G})$ are the maximal cliques of $\mathcal{G}$, $\mathcal{S}(\mathcal{G})$ the minimal complete separators, and $\nu_{\mathcal{G}}(S)$ are certain graph invariants.

How can posterior distributions of this form be represented and/or simulated and what are the properties of such distributions?

Even case where the graphs considered is all forests is difficult

Recent progress concerning *structural Markov properties* of distributions in (5) has been made by Byrne and Dawid (2015).

Summary for trees and forests

- *Direct likelihood methods* (ignoring penalty terms) lead to sensible results.
- (Boootstrap) sampling distribution of tree MLE can be *simulated*
- *Penalty terms additive along branches*, so highest AIC or BIC scoring tree (forest) also available using a MWST algorithm.
- Tree methods scale extremely well with both sample size and number of variables;
- Pairwise marginal counts are *sufficient statistics* for the tree problem (empirical covariance matrix in the Gaussian case).

Note sufficiency holds despite parameter space very different from open subset of $\mathcal{R}^k$.

Graphical lasso

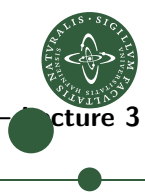
Consider an undirected Gaussian graphical model and the l_1-penalized log-likelihood function

$$2\ell_{pen}(K) = \log\det K - \mathrm{tr}(K\bar{W}) - \kappa||K||_1.$$

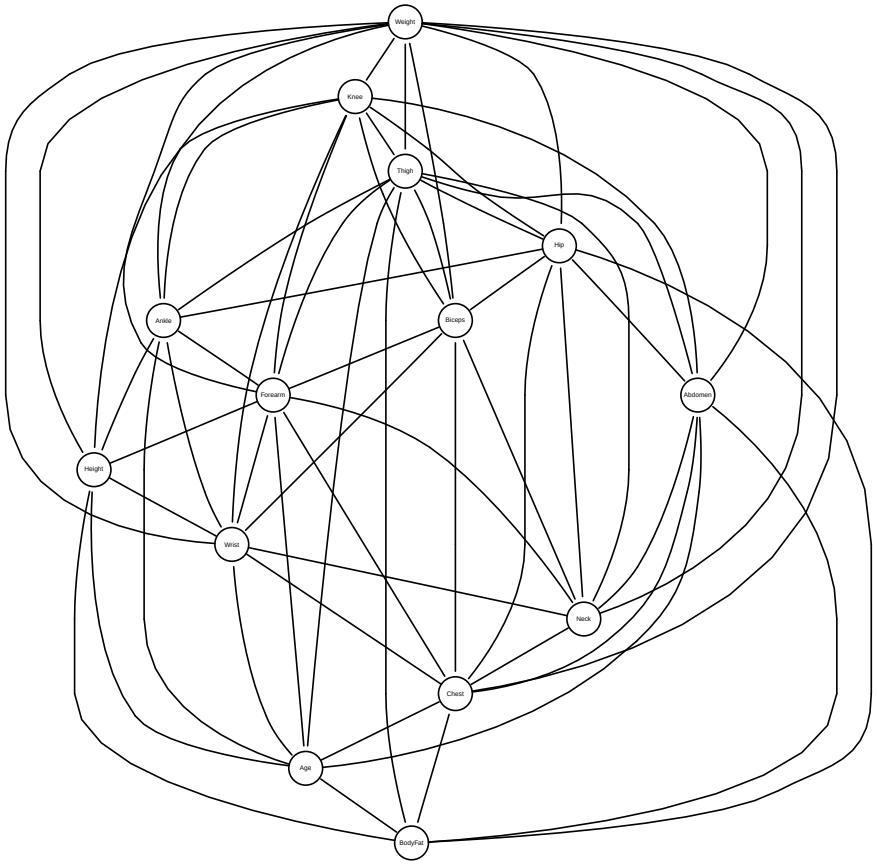
The penalty $||K||_1 = \sum_{u,v} |k_{uv}|$ is essentially a convex relaxation of the number of edges in the graph and optimization of the penalized likelihood will typically lead to several $k_{uv} = 0$ and thus in effect estimate a particular graph.

This penalized likelihood can be maximized efficiently (Banerjee et al., 2008) as implemented in the *graphical lasso* (Friedman et al., 2008).

Beware: not scale-invariant!



glasso for bodyfat



Optimizing the convex glasso problem

We shall maximize the penalized likelihood function

$$\ell(K) = \log\det(K) - \mathrm{tr}(\bar{W}K) - \kappa||K||_1.$$

Optimizing the convex glasso problem

We shall maximize the penalized likelihood function

$$\ell(K) = \log\det(K) - \mathrm{tr}(\bar{W}K) - \kappa||K||_1.$$

This has subgradient equation $\partial\ell(K) = 0$, where

$$\partial\ell(K) = \Sigma - \bar{W} - \kappa\Gamma$$

and $\Gamma = \mathrm{sign}(K)$ where $\mathrm{sign}(k_{uv}) = 1$ if $k_{uv} > 0$, $\mathrm{sign}(k_{uv}) = -1$ if $k_{uv} < 0$, and $\mathrm{sign}(k_{uv}) \in [-1, 1]$ if $k_{uv} = 0$.

Optimizing the convex glasso problem

We shall maximize the penalized likelihood function

$$\ell(K) = \log\det(K) - \mathrm{tr}(\bar{W}K) - \kappa||K||_1.$$

This has subgradient equation $\partial\ell(K) = 0$, where

$$\partial\ell(K) = \Sigma - \bar{W} - \kappa\Gamma$$

and $\Gamma = \mathrm{sign}(K)$ where $\mathrm{sign}(k_{uv}) = 1$ if $k_{uv} > 0$, $\mathrm{sign}(k_{uv}) = -1$ if $k_{uv} < 0$, and $\mathrm{sign}(k_{uv}) \in [-1, 1]$ if $k_{uv} = 0$.

Hence the glasso estimate $\check{\Sigma}$ of Σ satisfies

$$\check{\Sigma} = \bar{W} + \kappa\Gamma.$$

Compare to MLE

$$\hat{\Sigma} = \bar{W} + \Gamma^*$$

where $\gamma^*_{uv} = 0$ whenever $u \sim v$.

Blocking the subgradient equation

Write the subgradient equation in block matrix form with $S = \bar{W}$, the lower right corner being 1×1 we get

$$\begin{pmatrix} S_{11} & s_{12} \\ s_{12}^\top & s_{22} \end{pmatrix} - \begin{pmatrix} \Sigma_{11} & \sigma_{12} \\ \sigma_{12}^\top & \sigma_{22} \end{pmatrix} + \kappa \begin{pmatrix} \Gamma_{11} & \gamma_{12} \\ \gamma_{12}^\top & 1 \end{pmatrix} = 0.$$

Blocking the subgradient equation

Write the subgradient equation in block matrix form with $S = \bar{W}$, the lower right corner being 1×1 we get

$$\begin{pmatrix} S_{11} & s_{12} \\ s_{12}^\top & s_{22} \end{pmatrix} - \begin{pmatrix} \Sigma_{11} & \sigma_{12} \\ \sigma_{12}^\top & \sigma_{22} \end{pmatrix} + \kappa \begin{pmatrix} \Gamma_{11} & \gamma_{12} \\ \gamma_{12}^\top & 1 \end{pmatrix} = 0.$$

Focusing on the upper right block of this equation we get

$$s_{12} - \sigma_{12} + \kappa\gamma_{12} = 0.$$

Blocking the subgradient equation

Write the subgradient equation in block matrix form with $S = \bar{W}$, the lower right corner being 1×1 we get

$$\begin{pmatrix} S_{11} & s_{12} \\ s_{12}^\top & s_{22} \end{pmatrix} - \begin{pmatrix} \Sigma_{11} & \sigma_{12} \\ \sigma_{12}^\top & \sigma_{22} \end{pmatrix} + \kappa \begin{pmatrix} \Gamma_{11} & \gamma_{12} \\ \gamma_{12}^\top & 1 \end{pmatrix} = 0.$$

Focusing on the upper right block of this equation we get

$$s_{12} - \sigma_{12} + \kappa\gamma_{12} = 0.$$

Using the identity $(\Sigma_{11})^{-1}\sigma_{12} = -k_{22}^{-1}k_{12} = \beta$ and thus $\operatorname{sign}(k_{12}) = -\operatorname{sign}(\beta)$ we can rewrite this equation as

$$\Sigma_{11}\beta - s_{12} + \kappa \operatorname{sign}(\beta) = 0.$$

Lasso regression

The Lasso regression problem is

$$\text{minimize} \qquad (y - Z\beta)^\top (y - Z\beta)/2 + \kappa ||\beta||_1.$$

Lasso regression

The Lasso regression problem is

$$\text{minimize} \qquad (y - Z\beta)^\top (y - Z\beta)/2 + \kappa ||\beta||_1.$$

The subgradient equation for this problem becomes

$$Z^\top Z\beta - Z^\top y + \kappa \, \text{sign}(\beta) = 0.$$

Lasso regression

The Lasso regression problem is

$$\text{minimize} \qquad (y - Z\beta)^\top (y - Z\beta)/2 + \kappa ||\beta||_1.$$

The subgradient equation for this problem becomes

$$Z^\top Z\beta - Z^\top y + \kappa \operatorname{sign}(\beta) = 0.$$

Compare this to the subgradient equation for the graphical lasso

$$\Sigma_{11}\beta - s_{12} + \kappa \operatorname{sign}(\beta) = 0.$$

Lasso regression

The Lasso regression problem is

$$\text{minimize} \qquad (y - Z\beta)^\top (y - Z\beta)/2 + \kappa ||\beta||_1.$$

The subgradient equation for this problem becomes

$$Z^\top Z\beta - Z^\top y + \kappa \operatorname{sign}(\beta) = 0.$$

Compare this to the subgradient equation for the graphical lasso

$$\Sigma_{11}\beta - s_{12} + \kappa \operatorname{sign}(\beta) = 0.$$

There is a simple iterative cyclic descent algorithm for solving the first equation, and this can of course be used to solve the second equation.

Cyclic descent algorithm for graphical lasso

Define the *soft threshold function*

$$T(x,t) = \text{sign}(x)(|x| - t)_+;$$

Cyclic descent algorithm for graphical lasso

Define the *soft threshold function*

$$T(x, t) = \text{sign}(x)(|x| - t)_+;$$

Input: Empirical covariance matrix S; penalty parameter κ;

Cyclic descent algorithm for graphical lasso

Define the *soft threshold function*

$$T(x,t) = \text{sign}(x)(|x| - t)_+;$$

Input: Empirical covariance matrix S; penalty parameter κ;

Output: Glasso estimate $\hat{K}^\kappa$; concentration graph $\hat{\mathcal{G}}^\kappa$.

Cyclic descent algorithm for graphical lasso

Define the *soft threshold function*

$$T(x,t) = \mathrm{sign}(x)(|x| - t)_+;$$

Input: Empirical covariance matrix S; penalty parameter κ;

Output: Glasso estimate $\hat{K}^\kappa$; concentration graph $\hat{\mathcal{G}}^\kappa$.

1. **Initialize** $\Sigma \leftarrow S + \kappa I$; $\beta_{uv} \leftarrow 0, u, v \in V$.

Cyclic descent algorithm for graphical lasso

Define the *soft threshold function*

$$T(x,t) = \text{sign}(x)(|x| - t)_+;$$

Input: Empirical covariance matrix S; penalty parameter κ;

Output: Glasso estimate $\hat{K}^\kappa$; concentration graph $\hat{\mathcal{G}}^\kappa$.

1. **Initialize** $\Sigma \leftarrow S + \kappa I$; $\beta_{uv} \leftarrow 0, u, v \in V$.
2. **Repeat** for $v \in V$ **until** convergence

Cyclic descent algorithm for graphical lasso

Define the *soft threshold function*

$$T(x,t) = \text{sign}(x)(|x| - t)_+;$$

Input: Empirical covariance matrix S; penalty parameter κ;

Output: Glasso estimate $\hat{K}^\kappa$; concentration graph $\hat{\mathcal{G}}^\kappa$.

1. **Initialize** $\Sigma \leftarrow S + \kappa I$; $\beta_{uv} \leftarrow 0, u, v \in V$.
2. **Repeat** for $v \in V$ **until** convergence
   1. **For** $u \in V \setminus v$ **until convergence**:
      $$\beta_{uv} \leftarrow T\left(s_{uv} - \sum_{w \neq v} \sigma_{uw}\beta_{wv};\ \kappa\right)/\sigma_{vv};$$

Cyclic descent algorithm for graphical lasso

Define the *soft threshold function*

$$T(x,t) = \mathrm{sign}(x)(|x| - t)_+;$$

Input: Empirical covariance matrix S; penalty parameter κ;

Output: Glasso estimate $\hat{K}^\kappa$; concentration graph $\hat{\mathcal{G}}^\kappa$.

1. **Initialize** $\Sigma \leftarrow S + \kappa I$; $\beta_{uv} \leftarrow 0, u, v \in V$.
2. **Repeat** for $v \in V$ **until** convergence
   1. **For** $u \in V \setminus v$ **until convergence**:
      $\beta_{uv} \leftarrow T\left(s_{uv} - \sum_{w \neq v} \sigma_{uw}\beta_{wv};\ \kappa\right)/\sigma_{vv}$;
   2. **For** $u \in V \setminus \{v\}$ **do** $\sigma_{uv} \leftarrow \sum_{w \neq v} \sigma_{uw}\beta_{wv}$;

Cyclic descent algorithm for graphical lasso

Define the *soft threshold function*

$$T(x,t) = \text{sign}(x)(|x| - t)_+;$$

Input: Empirical covariance matrix S; penalty parameter κ;

Output: Glasso estimate $\hat{K}^\kappa$; concentration graph $\hat{\mathcal{G}}^\kappa$.

1. **Initialize** $\Sigma \leftarrow S + \kappa I$; $\beta_{uv} \leftarrow 0, u, v \in V$.
2. **Repeat** for $v \in V$ **until** convergence
   1. **For** $u \in V \setminus v$ **until convergence**:
      $$\beta_{uv} \leftarrow T\left(s_{uv} - \textstyle\sum_{w \neq v} \sigma_{uw}\beta_{wv};\ \kappa\right)/\sigma_{vv};$$
   2. **For** $u \in V \setminus \{v\}$ **do** $\sigma_{uv} \leftarrow \sum_{w \neq v} \sigma_{uw}\beta_{wv}$;
3. **For** $v \in V$ **do**:

Cyclic descent algorithm for graphical lasso

Define the *soft threshold function*

$$T(x,t) = \mathrm{sign}(x)(|x| - t)_+;$$

Input: Empirical covariance matrix S; penalty parameter κ;

Output: Glasso estimate $\hat{K}^\kappa$; concentration graph $\hat{\mathcal{G}}^\kappa$.

1. **Initialize** $\Sigma \leftarrow S + \kappa I$; $\beta_{uv} \leftarrow 0, u, v \in V$.
2. **Repeat** for $v \in V$ **until** convergence
   1. **For** $u \in V \setminus v$ **until convergence**:
      $\beta_{uv} \leftarrow T\left(s_{uv} - \sum_{w \neq v} \sigma_{uw}\beta_{wv};\ \kappa\right)/\sigma_{vv}$;
   2. **For** $u \in V \setminus \{v\}$ **do** $\sigma_{uv} \leftarrow \sum_{w \neq v} \sigma_{uw}\beta_{wv}$;
3. **For** $v \in V$ **do**:
   1. $\hat{k}_{vv} \leftarrow 1/(\sigma_{vv} - \sum_{w \neq v} \sigma_{vw}\beta_{wv})$

Cyclic descent algorithm for graphical lasso

Define the *soft threshold function*

$$T(x,t) = \text{sign}(x)(|x| - t)_+;$$

Input: Empirical covariance matrix S; penalty parameter κ;

Output: Glasso estimate $\hat{K}^\kappa$; concentration graph $\hat{\mathcal{G}}^\kappa$.

1. **Initialize** $\Sigma \leftarrow S + \kappa I$; $\beta_{uv} \leftarrow 0, u, v \in V$.
2. **Repeat** for $v \in V$ **until** convergence
   1. **For** $u \in V \setminus v$ **until convergence**:
      $\beta_{uv} \leftarrow T\left(s_{uv} - \sum_{w \neq v} \sigma_{uw}\beta_{wv};\ \kappa\right)/\sigma_{vv}$;
   2. **For** $u \in V \setminus \{v\}$ **do** $\sigma_{uv} \leftarrow \sum_{w \neq v} \sigma_{uw}\beta_{wv}$;
3. **For** $v \in V$ **do**:
   1. $\hat{k}_{vv} \leftarrow 1/(\sigma_{vv} - \sum_{w \neq v} \sigma_{vw}\beta_{wv})$
   2. **For** $u \in V \setminus v$ **do** $\hat{k}_{uv} \leftarrow -\beta_{uv} k_{vv}$.

Cyclic descent algorithm for graphical lasso

Define the *soft threshold function*

$$T(x,t) = \text{sign}(x)(|x| - t)_+;$$

Input: Empirical covariance matrix S; penalty parameter κ;

Output: Glasso estimate $\hat{K}^\kappa$; concentration graph $\hat{\mathcal{G}}^\kappa$.

1. **Initialize** $\Sigma \leftarrow S + \kappa I$; $\beta_{uv} \leftarrow 0, u, v \in V$.
2. **Repeat** for $v \in V$ **until** convergence
   1. **For** $u \in V \setminus v$ **until convergence**:
      $\beta_{uv} \leftarrow T\left(s_{uv} - \sum_{w \neq v} \sigma_{uw}\beta_{wv};\ \kappa\right)/\sigma_{vv}$;
   2. **For** $u \in V \setminus \{v\}$ **do** $\sigma_{uv} \leftarrow \sum_{w \neq v} \sigma_{uw}\beta_{wv}$;
3. **For** $v \in V$ **do**:
   1. $\hat{k}_{vv} \leftarrow 1/(\sigma_{vv} - \sum_{w \neq v} \sigma_{vw}\beta_{wv})$
   2. **For** $u \in V \setminus v$ **do** $\hat{k}_{uv} \leftarrow -\beta_{uv}k_{vv}$.
4. **Return** K and incidence graph of K.

An alternative algorithm

This algorithm updates 2×2 submatrices of K and resembles the IPS algorithm but also in some sense Kruskal's algorithm.

An alternative algorithm

This algorithm updates 2×2 submatrices of K and resembles the IPS algorithm but also in some sense Kruskal's algorithm.

Consider the restricted convex optimization problem:

$$\begin{aligned} &\text{minimize} && -\log\det(K) + \mathrm{tr}(KS) + \kappa||K||_1 \\ &\text{subject to} && k_{ij} = k^*_{ij} \text{ for } i \neq u \text{ or } j \neq v. \end{aligned}$$

An alternative algorithm

This algorithm updates 2×2 submatrices of K and resembles the IPS algorithm but also in some sense Kruskal's algorithm.

Consider the restricted convex optimization problem:

$$\begin{aligned} &\text{minimize} && -\log\det(K) + \mathrm{tr}(KS) + \kappa||K||_1 \\ &\text{subject to} && k_{ij} = k^*_{ij} \text{ for } i \neq u \text{ or } j \neq v. \end{aligned}$$

Using Schur complements, the objective function becomes equivalent to

$$-\log\det(K_{cc} - K_{ca}K_{aa}^{-1}K_{ac}) + \mathrm{tr}(K_{cc}S_{cc}) + \kappa||K_{cc}||_1$$

where $c = \{u, v\}$ and $a = V \setminus \{u, v\}$.

An alternative algorithm

This algorithm updates 2×2 submatrices of K and resembles the IPS algorithm but also in some sense Kruskal's algorithm.

Consider the restricted convex optimization problem:

$$\begin{aligned} &\text{minimize} && -\log\det(K) + \mathrm{tr}(KS) + \kappa||K||_1 \\ &\text{subject to} && k_{ij} = k^*_{ij} \text{ for } i \neq u \text{ or } j \neq v. \end{aligned}$$

Using Schur complements, the objective function becomes equivalent to

$$-\log\det(K_{cc} - K_{ca}K_{aa}^{-1}K_{ac}) + \mathrm{tr}(K_{cc}S_{cc}) + \kappa||K_{cc}||_1$$

where $c = \{u, v\}$ and $a = V \setminus \{u, v\}$.

This problem is trivial to solve without iteration.

An alternative algorithm

This algorithm updates 2×2 submatrices of K and resembles the IPS algorithm but also in some sense Kruskal's algorithm.

Consider the restricted convex optimization problem:

$$\begin{aligned} &\text{minimize} && -\log\det(K) + \text{tr}(KS) + \kappa||K||_1 \\ &\text{subject to} && k_{ij} = k^*_{ij} \text{ for } i \neq u \text{ or } j \neq v. \end{aligned}$$

Using Schur complements, the objective function becomes equivalent to

$$-\log\det(K_{cc} - K_{ca}K_{aa}^{-1}K_{ac}) + \text{tr}(K_{cc}S_{cc}) + \kappa||K_{cc}||_1$$

where $c = \{u, v\}$ and $a = V \setminus \{u, v\}$.

This problem is trivial to solve without iteration.

Iterating through edges in order of decreasing unexplained correlation should give a very efficient algorithm.

Banerjee, O., Ghaoui, L. E., and dAspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. 9:485–216.

Byrne, S. and Dawid, A. P. (2015). Structural Markov graph laws for Bayesian model uncertainty. *Annals of Statistics*, 43:1647.

Dawid, A. P. and Lauritzen, S. L. (1993). Hyper Markov laws in the statistical analysis of decomposable graphical models. *The Annals of Statistics*, 21:1272–1317.

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

Højsgaard, S., Edwards, D., and Lauritzen, S. (2012). *Graphical Models with R*. Springer-Verlag, New York.

Lauritzen, S. L. (1996). *Graphical Models*. Clarendon Press, Oxford, United Kingdom.