Fitting Gaussian graphical model with ggm

Giovanni Marchetti and Ilaria Carobbi

viale Morgagni, 59, 50134 Firenze, Italy E-mail: giovanni.marchetti@ds.unifi.it

1. Introduction

Graphical models are statistical models for data observed on a set of variables $Y_1, \dots, Y_d$, that specify a set of conditional and marginal independencies between the variables. The set of independencies is exactly determined by the structure of a graph having the variables as nodes.

2. Basic types of graph

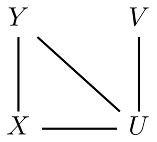
In general, a graph G is a mathematical object defined by a set of nodes (or vertices) V and by a set of edges E. In graphical models, graphs are used to define relation between the variables. Thus, the nodes in V are in 1-1 correspondence with the variables, and the edges in E are pairs of distinct nodes i and j denoting an association between the variables Y_i and Y_j. Edges can be undirected or directed. Undirected edges ij specify no order between i and j. They are typically denoted by a line $i — j$, or a dashed line, or by a bi-directed arrow $i \longleftrightarrow j$. Directed edges $i \to j$ specify an order with i coming before j. The graphs can be classified according to the type of edge they contain. The main types are:

- undirected graphs, containing only undirected edges;
- directed acyclic graphs, containing only directed edges;
- mixed graphs, containing both undirected and directed edges.

2.1. Undirected graphs

An undirected graph contains only undirected edges. In statistics it is suitable to describe the associations between variables that are considered *on equal standing.*

Consider the following example. Let Y, X, V, U be 4 variables and suppose that we define the undirected graph



with edges VU, YU, UX, YX. The graph may be defined by a square matrix, the *adjacency matrix*, that has the elements in positions (i, j) and (j, i) equal to 1 whenever the edge ij is in G. This can be defined in ggm using the constructor function UG:

```
> G = UG(~V * U + Y * U + U * X + Y * X)
> G

  V U Y X
V 0 1 0 0
U 1 0 1 1
Y 0 1 0 1
X 0 1 1 0
```

The output is a matrix whose row and column names are the nodes. The argument of the function UG is a *model formula* defining the edges of the graph by 4 two-way interaction terms. The same graph could be defined by a different model formula, with one two-factor interaction VU and a three-factor interaction YXU:

```
> G = UG(~V * U + Y * X * U)
> G
```

```
  V U Y X
V 0 1 0 0
U 1 0 1 1
Y 0 1 0 1
X 0 1 1 0
```

The two subgraphs defined by the subsets VU and YXU are complete, that is, they have all the possible edges:

```
> a = c("U", "V")
> b = c("Y", "X", "U")
> G[a, a]
```

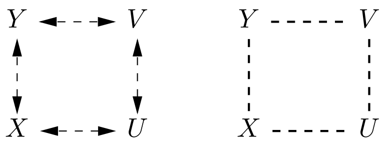
```
  U V
U 0 1
V 1 0
```

```
> G[b, b]
```

```
  Y X U
Y 0 1 1
X 1 0 1
U 1 1 0
```

Moreover, the two subgraphs are maximal in the sense that they cannot be enlarged without losing completeness. Therefore the two subsets VU and YXU are called two *cliques* of the graph. Finding the cliques of an undirected graph is an NP-hard problem.

Sometimes we will need undirected graphs with bi-directed or dashed edges like for example



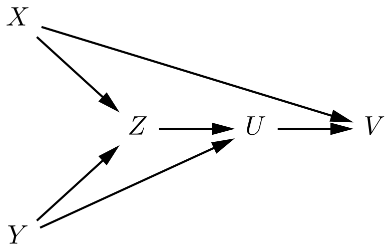
To distinguish these graphs we code in their adjacency matrix the edges by a 2 instead of a 1. Thus, with same function `UG`, we define the two previous graphs by

```
> G2 = 2 * UG(~Y * V + V * U + U * X + Y * X)
> G2
```

```
  Y V U X
Y 0 2 0 2
V 2 0 2 0
U 0 2 0 2
X 2 0 2 0
```

2.2. Directed acyclic graphs

A directed acyclic graph (a DAG for short) contains only directed edges. Each edge is a couple (i, j) of nodes, defining the tail and the head of an arrow $i \to j$. Moreover in a DAG there are no cycles, i.e., is impossible starting from a node and following the direction of the arrows to get back to the starting node. For example the following graph is a DAG.



In a DAG each node i has an associated set of *parents* meaning the (possibly empty) set pa(j) of the nodes i such that $i \to j$ is in G. Thus for example the parents of V are X and U, the parents of U are Z and Y, while X and Y have no parents each. A directed acyclic graph is used in statistics to specify a data generating process, where each variable is directly dependent by some parent variables, and indirectly dependent via intermediate variables.

A DAG G may be defined by its *adjacency matrix*, a square matrix that has the elements in position (i, j) equal to 1 whenever the edge $i \to j$ is in G. In ggm the previous DAG is defined by a constructor function DAG that takes as arguments several model formulae giving the parents of each node (except, possibly, the nodes with no parents). The previous DAG is defined as follows:

```
> D = DAG(V ~ X + U, U ~ Z + Y, Z ~ X + Y)
> D
```

```
  V X U Z Y
V 0 0 0 0 0
X 1 0 0 1 0
U 1 0 0 0 0
Z 0 0 1 0 0
Y 0 0 1 1 0
```

A property of DAGs is that the nodes can be always reordered (not uniquely, in general) such that the parents are before the children. This is called the *topological order* of the DAG. For instance

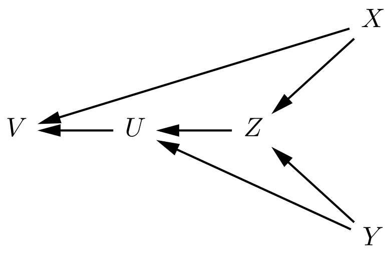
```
> Do = topSort(D)
> Do
```

```
  Y X Z U V
Y 0 0 1 1 0
X 0 0 1 0 1
Z 0 0 0 1 0
U 0 0 0 0 1
V 0 0 0 0 0
```

produces the adjacency matrix in the topological order, that turns out to be always upper triangular.

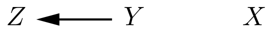
Drawing graphs from right to left

The description of a DAG by a sequence of model formulae implies that each formula defines a node and its parents and that the parents are to the right of the symbol $\sim$. Even if this is the opposite of the topological order, sometimes it is convenient and therefore we shall draw a DAG from right to left. For example, the DAG defined above could be drawn as follows.



Graphs with isolated nodes

A node is isolated if the set of its parents and children are empty. A DAG with an isolated node X can be defined by introducing in the model formula a term $X \sim X$. For instance the graph



is defined by

```
> D = DAG(Z ~ Y, X ~ X)
> D
```

```
  Z Y X
Z 0 0 0
Y 1 0 0
X 0 0 0
```

3. Some basic Gaussian graphical models

We distinguish 5 types of graphical models

1. Covariance graph models
2. Concentration graph models
3. Regression graph models
4. General multivariate regression graph models
5. Linear structural equation models

These models can be developed for both continuous or discrete data, but our discussion will be limited to the first situation.

In this case, the standard assumption is that the observed data are a random sample from a d-dimensional Gaussian distribution $Y = (Y_1, \dots, Y_d) \sim N(\mu, \Sigma)$, representing the population, where μ is the mean vector that, without loss of generality, can be assumed to be zero, and where $\Sigma = [\sigma_{ij}]$ is a p.d. covariance matrix. Often the Gaussian assumption is too strong, but as in linear regression, we can fit the models with a set weaker assumptions provided that we include appropriate nonlinear terms (see the package **checklin**). Each model can be simply characterized by a set of linear constraints on parameters of the population.

3.1. Covariance graph models

A Gaussian covariance graph model for Y is defined by zero constraints on the covariances as specified by the missing edges of an undirected graph G. The model specifies that

$$\sigma_{ij} = 0 \text{ whenever the edge } ij \text{ is not in } G.$$

In a Gaussian distribution $\sigma_{ij} = 0$ if and only the two variables Y_i and Y_j are marginally independent. Therefore a covariance graph model specifies an independence $Y_i \perp\!\!\!\perp Y_j$ for each missing edge ij. The model belongs to the class of linear in covariance structures (cf. Anderson, 1971).

Example

We analyze for a sample of 72 students, the covariance matrix among 4 variables measuring 4 different strategies to cope with stress. The variables are Y, cognitive avoidance; X, vigilance; V, blunting and U, monitoring. The data are contained in the package `SIN`. The following instructions compute the sample covariance matrix.

```
> require(SIN)
> data(stressful)
> S = sdcor2cov(stressful$stddev, stressful$corr)
> dimnames(S) = list(c("Y", "X", "V", "U"), c("Y", "X", "V", "U"))
> S
```

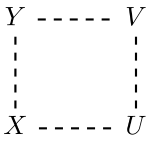
```
          Y         X         V         U
Y 45.832900 -8.652060  6.602104  0.207839
X -8.652060 40.832100  0.000000  9.220131
V  6.602104  0.000000  4.494400 -0.976260
U  0.207839  9.220131 -0.976260  9.424900
```

Inspection of the sample correlation matrix

```
> cov2cor(S)
```

```
      Y     X     V     U
Y  1.00 -0.20  0.46  0.01
X -0.20  1.00  0.00  0.47
V  0.46  0.00  1.00 -0.15
U  0.01  0.47 -0.15  1.00
```

shows that a possible hypothesis to test is $\sigma_{XV} = \sigma_{YU} = 0$. This corresponds to a covariance graph model



This model can be fitted by maximum likelihood with

```
> G <- 2 * UG(~Y * X + X * U + U * V + V * Y)
> G
```

```
  Y X U V
Y 0 2 0 2
X 2 0 2 0
U 0 2 0 2
V 2 0 2 0
```

```
> fitCovGraph(G, S, n = 72)
```

```
$Shat
          Y         X         V         U
Y 45.810113 -8.816792  6.617870  0.000000
X -8.816792 40.915862  0.000000  9.254775
V  6.617870  0.000000  4.501194 -1.000127
U  0.000000  9.254775 -1.000127  9.421285

$dev
[1] 0.007675671

$df
[1] 2
```

```
$it
[1] 8
```

The output of the function is a list with components `Shat`, the fitted covariance matrix, `dev`, the likelihood ratio statistic against the saturated model, i.e., the model with no restrictions on the covariances, and `it`, the number of iterations of the algorithm. The fit for the `stressful` data is almost perfect.

3.2. Concentration graph models

A Gaussian concentration graph model for a random vector Y is defined by zero constraints on the inverse of the covariance matrix Σ. This is called the *concentration matrix* and is denoted by $\Sigma^{-1} = [\sigma^{ij}]$. Given an undirected graph G this model specifies that

$$\sigma^{ij} = 0 \text{ whenever the edge } ij \text{ is not in } G.$$

If Y has a Gaussian distribution, the constraint $\sigma^{ij} = 0$ is shown to be equivalent to the *conditional independence* $Y_i \perp\!\!\!\perp Y_j|$ given all the remaining variables. Thus the concentration graph model specifies a set of pairwise conditional independencies for each missing edge in the graph.

undirected graphs G is defined by a set of nodes, usually denoted by integers $i = 1, \dots, d$, and by a set E of edges, described by pairs $\{i, j\}$ of nodes. The *set of missing edges* is the complementary set of E. A simple representation of the graph is given by its *adjacency matrix*. A concentration graph model is also called a *covariance selection model*; see Dempster (1972).

A Gaussian undirected graph model for a d-dimensional random vector $Y = (Y_1, \dots, Y_d)$ is defined by a family of Gaussian distributions with mean vector zero and a $d \times d$ covariance matrix $\Sigma = [\sigma_{ij}]$ such that its inverse $\Sigma^{-1} = [\sigma^{ij}]$ satisfies

$$\sigma^{ij} = 0 \text{ whenever } i \text{ — } j \text{ is not in } G$$

The inverse of the covariance matrix is called a *concentration matrix* and therefore the model specifies zero constraints on the concentrations. The model is sometimes called a *covariance selection model* (Dempster) or *concentration graph model* (Cox and Wermuth, 1996).

The concentration graph model specifies a set of conditional independencies: for each *missing* edge i — j, the variables Y_i and Y_j are independent given all the remaining variables $Y_{V\setminus ij}$.

example

We analyze a covariance matrix for a sample of 684 female college students, concerning 4 variables on anxiety and anger. The 4 variables are: Y = state anxiety, X = state anger, V = trait anxiety, U = trait anger. The sample covariance matrix is as follows

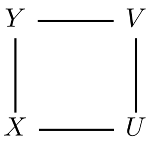
```
> S = matrix(c(37.1926, 24.9311, 21.6056, 15.6907, 24.9311, 44.8472,
+     17.8072, 21.5865, 21.6056, 17.8072, 32.2462, 18.3523, 15.6907,
+     21.5865, 18.3523, 43.1191), 4, 4)
> dimnames(S) = list(c("Y", "X", "V", "U"), c("Y", "X", "V", "U"))
> S

        Y       X       V       U
Y 37.1926 24.9311 21.6056 15.6907
X 24.9311 44.8472 17.8072 21.5865
V 21.6056 17.8072 32.2462 18.3523
U 15.6907 21.5865 18.3523 43.1191
```

A preliminary analysis is that of computing the marginal and partial correlation matrices. The function `correlations` stores the marginal and partial correlations in the lower and upper triangle of a matrix, respectively:

```
> correlations(S)
```

```
          Y         X          V           U
Y 1.0000000 0.4485459 0.46954702 -0.03769731
X 0.6104435 1.0000000 0.02522645  0.31929907
V 0.6238767 0.4682618 1.00000000  0.31833491
U 0.3918133 0.4908850 0.49217141  1.00000000
```

The almost zero partial correlations $\hat{\rho}_{YU.XV}$ and $\hat{\rho}_{XV.YU}$ suggest the concentration graph model



This can be fitted by ML using the function `fitConGraph` as follows.

```
> G = UG(~Y * V + V * U + U * X + X * Y)
> ml = fitConGraph(amat = G, S, n = 684)
> ml
```

```
$Shat
         Y        X        V        U
Y 37.19260 24.93110 21.60560 16.79689
X 24.93110 44.84720 16.98069 21.58650
V 21.60560 16.98069 32.24620 18.35230
U 16.79689 21.58650 18.35230 43.11910

$dev
[1] 1.947154

$df
[1] 2

$it
[1] 7
```

The output is the same of the function `fitConGraph` where `Shat` is the fitted covariance matrix $\hat{\Sigma} = [\hat{\sigma}_{ij}]$. The fitted concentration matrix is simply the inverse $\hat{\Sigma}^{-1}$ and we can verify that the two conditions of the maximum likelihood estimates are verified, that is (a) the concentrations corresponding to the missing edges are zero:

```
> round(solve(ml$Shat), 4)
```

```
        Y       X       V       U
Y  0.0568 -0.0214 -0.0268  0.0000
X -0.0214  0.0398  0.0000 -0.0116
V -0.0268  0.0000  0.0568 -0.0137
U  0.0000 -0.0116 -0.0137  0.0348
```

and (b) the fitted covariances corresponding to the edges present coincide with the sample covariances.

3.3. Directed acyclic graph models

A Gaussian directed acyclic graph model for a d-dimensional random vector Y is defined by the recursive equations

$$Y = BY + \epsilon, \text{ where } \epsilon \sim N(0, \Delta)$$

is a vector of independent residuals with a diagonal covariance matrix Δ and $B = [\beta_{ij}]$ is a matrix of coefficients such that

$$\beta_{ij} = 0 \text{ whenever } i \leftarrow j \text{ is not in } G.$$

Therefore the concentration and covariance matrices of Y are

$$\Sigma^{-1} = A^T \Delta^{-1} A, \quad \Sigma = A^{-1} \Delta A^T$$

where $A = I - B$.

The model specifies a set of conditional independencies between Y_i and Y_j given all the remaining variables $Y_{V \setminus ij}$ if the edge $\{i, j\}$ is missing.

3.4. Structural equation models