Advanced techniques in applied economics

Lecture 2: Gaussian and non-paranormal graphical models

Piotr Zwiernik

upf. Universitat Pompeu Fabra Barcelona

Statistics, Probability and Machine Learning Research Group

BSE Barcelona School of Economics

Spring 2026

From conditional independence to networks

What did we learn last time?

Last time we introduced conditional independence. In particular, for Gaussian data,

$$X_i \perp\!\!\!\perp X_j \mid X_{\setminus ij} \qquad \Longleftrightarrow \qquad K_{ij} = 0,$$

where $K = \Sigma^{-1}$ is the precision matrix.

Today we encode these conditional independence statements by **graphs**. We show how to interpret them and then learn those graphs from data.

Roadmap for today

1. Undirected graphs and conditional independence
2. Gaussian graphical models
3. Likelihood and estimation under graph constraints
4. High-dimensional learning and the graphical lasso
5. Robustness: non-paranormal graphical models
6. Application and interpretation

Part 1: Undirected graphical models

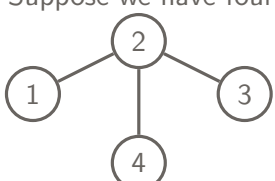
Undirected graphical models

Let $V = \{1, \ldots, m\}$ index variables $X = (X_1, \ldots, X_m)$. An **undirected graph** $G = (V, E)$ has

- one vertex for each variable,
- an edge $i - j$ if the model does *not* impose $X_i \perp\!\!\!\perp X_j \mid X_{V \setminus \{i,j\}}$.

We say that **X** is **pairwise Markov with respect to** the graph **G**.

Example: Suppose we have four variables X_1, X_2, X_3, X_4 and the graph



$$X_1 \perp\!\!\!\perp X_3 \mid X_2, X_4$$

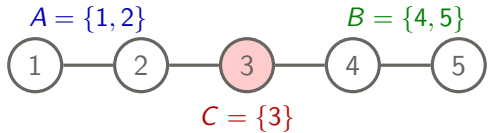
$$X_1 \perp\!\!\!\perp X_4 \mid X_2, X_3$$

$$X_3 \perp\!\!\!\perp X_4 \mid X_1, X_2$$

For a standard introduction to undirected graphical models, see Chapter 3 in Lauritzen, *Graphical Models*. A more applied introduction with R examples is Højsgaard, Edwards, and Lauritzen, *Graphical Models with R*.

Global Markov property

Graph separation: Let A, B, and C be disjoint subsets of vertices. If every path from A to B passes through C, then C **separates** A and B.



We say that $\mathbf{X}$ satisfies the **global Markov property** with respect to G if

$$C \text{ separates } A \text{ and } B \quad \Longrightarrow \quad X_A \perp\!\!\!\perp X_B \mid X_C.$$

In the graph above,

$$(X_1, X_2) \perp\!\!\!\perp (X_4, X_5) \mid X_3.$$

Clearly the global Markov property implies the pairwise Markov property.

For precise definitions see Sections 3.1–3.3 in Lauritzen, *Graphical Models*.

Hammersley–Clifford theorem

Graph factorization: If $C_1, \ldots, C_r$ are the cliques of G, then

$$f(x) \propto \prod_{k=1}^{r} \psi_{C_k}(x_{C_k})$$

for suitable clique potentials ψ_{C_k}.

Hammersley–Clifford Theorem

For a **positive** distribution (e.g. Gaussian), the following are equivalent:

1. the density factorizes over the cliques of G.
2. the distribution is pairwise Markov with respect to the graph G,
3. the distribution is global Markov with respect to the graph G,

Hammersley–Clifford connects graph separation with probabilistic factorization.

For the Hammersley–Clifford theorem and its role in graphical model theory, see Section 3.2 in Lauritzen, *Graphical Models*.

A tiny factorization example

Consider the chain graph

Its cliques are $\{1, 2\}$ and $\{2, 3\}$, so a positive graphical model over this graph has density

$$f(x_1, x_2, x_3) \propto \psi_{12}(x_1, x_2)\, \psi_{23}(x_2, x_3).$$

Why this implies conditional independence

Fixing x_2, we get

$$f(x_1, x_3 \mid x_2) = \frac{\psi_{12}(x_1, x_2)}{\sum_{x_1} \psi_{12}(x_1, x_2)} \, \frac{\psi_{23}(x_2, x_3)}{\sum_{x_3} \psi_{23}(x_2, x_3)}.$$

The first factor depends only on x_1, the second only on x_3, so

$$f(x_1, x_3 \mid x_2) = f(x_1 \mid x_2)\, f(x_3 \mid x_2).$$

Hence $X_1 \perp\!\!\!\perp X_3 \mid X_2$.

Part 2: Gaussian graphical models

Gaussian graphical models

Let $X \sim \mathcal{N}_m(\mu, \Sigma)$ and let $K = \Sigma^{-1}$. Given an undirected graph $G = (V, E)$, the corresponding **Gaussian graphical model** is

$$\mathcal{M}(G) = \left\{\mathcal{N}_m(\mu, \Sigma) : K_{ij} = 0 \text{ whenever } i \not\sim j\right\}.$$

For Gaussian data,

$$i \not\sim j \iff K_{ij} = 0 \iff X_i \perp\!\!\!\perp X_j \mid X_{V\setminus\{i,j\}}.$$

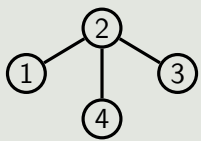
Learning the graph is the same as learning the zero pattern of the precision matrix.

For Gaussian graphical models and the link between zeros in the precision matrix and conditional independence, see Chapter 5 in Lauritzen, *Graphical Models*.

A four-variable example

Consider the following model and its corresponding graph

$$K = \begin{pmatrix} * & * & 0 & 0 \\ * & * & * & * \\ 0 & * & * & 0 \\ 0 & * & 0 & * \end{pmatrix}$$



The zeros $K_{13} = K_{14} = K_{34} = 0$ mean

$$X_1 \perp\!\!\!\perp X_3 \mid X_{\{2,4\}}, \qquad X_1 \perp\!\!\!\perp X_4 \mid X_{\{2,3\}}, \qquad X_3 \perp\!\!\!\perp X_4 \mid X_{\{1,2\}}.$$

Hammersley–Clifford then gives, for example, $X_1 \perp\!\!\!\perp X_3 \mid X_2$.

The precision matrix describes what remains after controlling for the other variables.

Part 3: Likelihood and estimation

The Gaussian likelihood

Suppose $X^{(1)}, \dots, X^{(n)} \overset{iid}{\sim} \mathcal{N}_m(\mu, \Sigma)$ and let

$$S_n = \tfrac{1}{n} \textstyle\sum_{k=1}^{n} (X^{(k)} - \bar{X}_n)(X^{(k)} - \bar{X}_n)^\top$$

be the sample covariance matrix.

Maximizing the Gaussian log-likelihood over μ gives $\widehat{\mu} = \bar{X}_n$. Plugging this back in yields the profile likelihood

$$\tilde{\ell}_n(K) \;:=\; \frac{2}{n}\, \ell_n(\bar{X}_n, K^{-1}) \;=\; \log\det K - \operatorname{tr}(S_n K)$$

up to an additive constant.

If there are no graph constraints and S_n is invertible, then $\hat{K} = S_n^{-1}$.

Estimation under a fixed graph

If the graph G is known, we maximize the Gaussian likelihood subject to the zero constraints

$$K_{ij} = 0 \qquad \text{for all } i \not\sim j.$$

Constrained MLE

$$\hat{K}_G = \arg\max \Big\{ \log\det K - \operatorname{tr}(S_n K) \Big\} \quad \text{over } K \in \mathbb{S}_+^m \text{ such that } K_{ij} = 0 \text{ if } i \not\sim j.$$

Why this is well behaved

The objective is strictly concave in K, and the constraints are linear. So whenever an optimum exists, it is uniquely defined.

For the convex-optimization viewpoint, see Section 3.1.5 in Boyd& Vandenberghe. For Gaussian likelihood inference under graphical constraints, see Chapters 5 and 6 in Lauritzen, *Graphical Models*.

Likelihood equations under a fixed graph

Recall the profile log-likelihood $\tilde{\ell}_n(K) = \log\det K - \operatorname{tr}(S_n K)$, $K \in \mathbb{S}^m_+ \cap V(G)$, where

$$V(G) = \{K \in \mathbb{S}^m : \ K_{ij} = 0 \text{ whenever } i \not\sim j\}.$$

The **gradient** with respect to K is: $\quad \nabla\tilde{\ell}_n(K) = K^{-1} - S_n = \Sigma - S_n.$

First-order conditions

At the MLE $\widehat{K}_G$, the gradient must be orthogonal to the linear space $V(G)$. Equivalently,

$$\widehat{\Sigma}_G - S_n \in V(G)^\perp.$$

So $\qquad (\widehat{\Sigma}_G)_{ij} = (S_n)_{ij} \qquad$ for all $i = j$ or $i \sim j$.

The MLE matches the sample covariance on the diagonal and on the edges.

For covariance completion in Gaussian graphical models, see Section 5.3 in Lauritzen, *Graphical Models*.

What is hard?

Even for Gaussian data, two different problems must be separated.

1. **Estimation:** if the graph is known, estimate K under the zero constraints.
2. **Structure learning:** if the graph is unknown, search over many possible zero patterns.

Combinatorial explosion

The number of undirected graphs on m vertices is

$$2^{\binom{m}{2}}.$$

So exhaustive search is impossible even for moderate m.

The real challenge is to learn the sparsity pattern.

A quick concentration experiment

Even in a simple Gaussian model, estimating a large covariance matrix is difficult.

```
library(MASS)
set.seed(1)
m <- 100
n <- 80
Sigma <- diag(m)
X <- mvrnorm(n, mu = rep(0,m), Sigma = Sigma)
S <- cov(X)
range(eigen(S)$values)
det(S) # numerically unstable or zero when m >= n

> range(eigen(S)$values)
[1] -5.195141e-16 4.485742e+00
```

Problem

The sample covariance matrix may be poorly behaved exactly in the regimes where network methods are most needed.

On concentration of the covariance matrix

```
> K <- matrix(c(1,0,1/2,0,1,1/2,1/2,1/2,1),3,3); Sig <- solve(K)
> set.seed(1)
> X10 <- mvrnorm(10,c(0,0,0),Sig); S10 <- cov(X10)
> X100 <- mvrnorm(100,c(0,0,0),Sig); S100 <- cov(X100)
> X1000 <- mvrnorm(1000,c(0,0,0),Sig); S1000 <- cov(X1000)
> solve(S10); solve(S100); solve(S1000)
          [,1] [,2] [,3]
[1,] 1.9379597 1.616762 0.8595338
[2,] 1.6167622 4.076235 2.1360457
[3,] 0.8595338 2.136046 1.6042403
           [,1] [,2] [,3]
[1,] 1.04792019 0.08406772 0.5781926
[2,] 0.08406772 0.93313405 0.3880432
[3,] 0.57819258 0.38804318 1.1148584
            [,1] [,2] [,3]
[1,] 0.88842341 -0.02029737 0.4710935
[2,] -0.02029737 0.90492134 0.4558757
[3,] 0.47109348 0.45587568 0.9628081
```

Without regularization estimating a covariance matrix is a hard problem.

Estimating the graph seems easier! (at least in this favorable case)

Why naive covariance inversion fails

When the dimension m is large relative to the sample size n:

- S_n is noisy
- S_n^{-1} is unstable
- if $m > n$, then S_n is singular.

This is the same basic problem as in high-dimensional regression.

Regression

Estimate $\beta \in \mathbb{R}^p$ (p large) in

$$Y = X\beta + \varepsilon.$$

Solution: impose sparsity with LASSO.

Graph learning

Estimate the precision matrix $K = \Sigma^{-1} \in \mathbb{S}_+^m$.
When m is large, we impose sparsity on K.

Graphical LASSO is the precision-matrix analogue of LASSO regression.

Graphical lasso

A standard approach is to add an ℓ_1 penalty to encourage sparsity:

$$\hat{K}_\lambda = \arg\min_{K \in \mathbb{S}_+^m} \Big\{ -\log\det K + \operatorname{tr}(S_n K) + \lambda \sum_{i \neq j} |K_{ij}| \Big\}.$$

The role of the penalty parameter λ

The penalty shrinks many off-diagonal entries toward zero, producing a sparse estimated graph.

Why the ℓ_1 penalty helps?

Economic reason

A sparse graph encodes the belief that most pairs of variables are not directly linked once the rest is controlled for.

Statistical reason

It reduces variance and stabilizes estimation when m is large.

Interpretive reason

It produces sparse networks that can actually be read and discussed.

The graphical lasso was introduced in Friedman, Hastie, and Tibshirani (2008), *Biostatistics*. For high-dimensional consistency results, see Ravikumar et al. (2011), *Annals of Statistics*.

A useful dual view of the graphical lasso

Dual formulation

The dual variable is the covariance matrix $\Sigma = K^{-1}$, and the problem becomes

$$\widehat{\Sigma}_\lambda = \arg\max_{\Sigma \in \mathbb{S}_+^m} \{\log\det \Sigma\} \quad \text{subject to} \quad \Sigma_{ii} = S_{ii}, \qquad |\Sigma_{ij} - S_{ij}| \le \lambda \;\; (i \neq j).$$

We optimize over covariance matrices Σ that stay inside an entrywise ℓ_∞-box around the sample covariance S_n. This dual viewpoint underlies many graphical lasso algorithms.

A simple block coordinate descent algorithm

The diagonal of $\widehat{\Sigma}$ is fixed. Starting at any feasible point Σ, we will update the i-th column $\Sigma_{\setminus i,i}$.

Partition the dual variable and the sample covariance as

$$\Sigma = \begin{bmatrix} \Sigma_{ii} & \Sigma_{i,\setminus i} \\ \Sigma_{\setminus i,i} & \Sigma_{\setminus i,\setminus i} \end{bmatrix}, \qquad S_n = \begin{bmatrix} S_{ii} & S_{i,\setminus i} \\ S_{\setminus i,i}^\top & S_{\setminus i,\setminus i} \end{bmatrix},$$

and hold everything but $z := \Sigma_{\setminus i,i}$ fixed. Denote $A = \Sigma_{\setminus i,\setminus i} \in \mathbb{S}_+^{m-1}$.

Blockwise optimization leads to a quadratic program

Use the Schur complement

Since

$$\det(\Sigma) = \det(A)\big(\Sigma_{11} - z^\top A^{-1} z\big),$$

maximizing $\det(\Sigma)$ is equivalent to

$$\min_z \quad z^\top A^{-1} z \qquad \text{subject to} \qquad \|z - S_{\setminus i,i}\|_\infty \le \lambda.$$

This is a simple convex quadratic program.

Algorithmic takeaway

Cycle through the rows/columns, solve one QP at a time, and update until convergence. This is exactly the kind of primal–dual block-coordinate idea that also drives GOLAZO.

To understand better the convex optimization story, see Section 8 in Lauritzen, Zwiernik, Locally associated graphical models and mixed convex exponential families, Annals of Statistics, 2022.

Choosing the penalty parameter

In practice, λ must be chosen from the data. Common approaches include:

- cross-validation,
- information criteria such as BIC or EBIC,
- stability-based methods.

Practical lesson

There is no single "correct" graph. What we estimate depends on the statistical criterion and the amount of regularization.

EBIC-based choices are convenient and are implemented in packages such as `qgraph` and `huge`.

For EBIC-type graph selection in Gaussian graphical models, see Foygel and Drton (2010), *Extended Bayesian Information Criteria for Gaussian Graphical Models*.

A first Gaussian graph in R

Example data

The dataset `stockdata` contains stock-return data. Here we take the first 40 variables, standardize them, and estimate a sparse Gaussian graphical model.

```
library(huge)
library(qgraph)
data(stockdata)
x <- scale(stockdata$data[,1:40])
out <- huge(x, method = "glasso")
plot(out)
```

```
# pick one graph along the regularization path
Khat <- out$icov[[10]]
Khat <- (Khat + t(Khat))/2

qgraph(-cov2cor(Khat),
       layout = "spring")
```

What is the path?

`huge(..., method = "glasso")` fits the graphical LASSO for many values of the penalty parameter λ. This gives a whole **regularization path**, from denser graphs to sparser ones. The command `out$icov[[10]]` picks one particular estimate along that path.

For the `huge` package, see Zhao et al. (2014), *Journal of Statistical Software*. For visualization with `qgraph`, see Epskamp et al. (2012), *Journal of Statistical Software*.

Part 5: Beyond Gaussianity

Why Gaussianity is often too optimistic

Economic data are rarely exactly Gaussian.

- asset returns are heavy-tailed,
- macro and firm-level variables may be skewed,
- relationships are often nonlinear,

We would like a model that keeps the Gaussian graphical machinery, but is less sensitive to non-Gaussian marginals.

Elliptical distributions "graphical models"

One idea is to learn the partial correlation graph for elliptical distributions (the same shape as Gaussian but potentially heavier tails; e.g. multivariate t-distribution.

See e.g. Finegold, Drton, Robust Graphical Modeling with t-Distributions, UAI, 2009. For the interpretation of such graphical model see Rossell, Zwiernik, Dependence in elliptical partial correlation graphs, EJS, 2021.

Gaussian copulas and the non-paranormal model

A flexible alternative is to assume that the observed variables are monotone transformations of latent Gaussian variables.

Non-paranormal model

We say that $X = (X_1, \ldots, X_m)$ is **non-paranormal** if there exist strictly monotone functions $f_1, \ldots, f_m$ such that

$$(Y_1, \ldots, Y_m) := (f_1(X_1), \ldots, f_m(X_m)) \sim \mathcal{N}_m(\mu, \Sigma).$$

Key fact

Such transformations preserve conditional independence: $X_i \perp\!\!\!\perp X_j \mid X_{\setminus ij} \iff Y_i \perp\!\!\!\perp Y_j \mid Y_{\setminus ij}$.

The non-paranormal model was introduced in Liu, Lafferty, and Wasserman (2009). For rank-based estimation in Gaussian copula graphical models, see Liu et al. (2012), *Annals of Statistics*.

Kendall's tau and monotone invariance

For two real-valued variables X_i and X_j, Kendall's tau is defined by

$$\tau_{ij} = \mathbb{P}\big((X_i - \widetilde{X}_i)(X_j - \widetilde{X}_j) > 0\big) - \mathbb{P}\big((X_i - \widetilde{X}_i)(X_j - \widetilde{X}_j) < 0\big),$$

where $(\widetilde{X}_i, \widetilde{X}_j)$ is an independent copy of (X_i, X_j).

Kendall's tau is a rank-based measure of dependence. It compares how often two observations move in the same order (concordant pairs) versus opposite order (discordant pairs).

Crucial properties

1. Kendall's tau is invariant under strictly monotone transformations:

$$\tau(Y_i, Y_j) \;=\; \tau(f_i(X_i), f_j(X_j)) \;=\; \tau(X_i, X_j)$$

and so it can be estimated from the observed data X.

2. If (Y_i, Y_j) is bivariate Gaussian with correlation ρ_{ij}, then

$$\tau_{ij} = \tau(Y_i, Y_j) = \frac{2}{\pi}\arcsin(\rho_{ij}), \qquad \text{equivalently} \qquad \rho_{ij} = \sin\Big(\frac{\pi}{2}\tau_{ij}\Big).$$

Kendall's tau as a bridge to latent Gaussian correlation

This gives a clear procedure

1. Use observations of X to estimate $\tau(X_i, X_j) = \tau(Y_i, Y_j)$; denote the estimator by $\widehat{\tau}_{ij}$.

2. Obtain the plug-in estimator of $\rho_{ij} = \text{cor}(Y_i, Y_j)$ via

$$\widehat{\rho}_{ij} = \sin(\frac{\pi}{2}\widehat{\tau}_{ij}).$$

3. $\widehat{\Sigma} = [\widehat{\rho}_{ij}]$ is an estimator of the correlation matrix of a Gaussian vector Y. We can plug it to the log-likelihood function and learn the graph using the graphical LASSO approach.

This gives a way to estimate the latent Gaussian correlation matrix from the observed data **without knowing the transformations** f_i. Beautiful!

For the relation between Kendall's tau and Gaussian copula correlation, see Section 3 in Liu et al. (2012).

Non-paranormal estimation in R

We can use the huge package again with a simple skeptic modification.

```
library(huge)
data(stockdata)

x <- scale(stockdata$data[,1:40])

# rank-based transformation
x.npn <- huge.npn(x,
                  npn.func = "skeptic")
```

```
out.npn <- huge(x.npn,
                method = "glasso")

Khat <- out.npn$icov[[10]]
Khat <- (Khat + t(Khat))/2

qgraph(-cov2cor(Khat),
       layout = "spring")
```

Interpretation

This estimates a sparse graph after a rank-based non-paranormal correction.

This approach can be extended to elliptical copulas, given an even wider scope. The problem is that we loose the conditional independence interpretation.

Part 6: Interpreting estimated networks

Why network summaries can be useful

Suppose we estimate a graph $\widehat{G}$ from data using a procedure such as the graphical LASSO.

In small examples we can inspect the graph directly. In modern applications, however, the graph is often far too large to read node by node.

Main point

In high-dimensional applications we rarely learn much from staring at the whole graph. What is often more useful are **higher-order summaries** of the estimated network.

These summaries can help us:

- identify variables that are locally central,
- detect tightly connected groups of variables,
- compare how sparse or fragmented different estimated graphs are,
- summarize dependence patterns in a way that is easier to report and interpret.

Examples from applied economics

Depending on the application, we may be interested in questions such as:

- **Financial returns:** which firms or sectors are most central after controlling for the rest?
- **Macroeconomic indicators:** do inflation, labor, credit, and production variables form separate communities?
- **Input-output or trade data:** are there blocks of industries or countries that remain strongly connected conditionally on the rest?
- **Consumer behavior:** which products or spending categories form tight clusters in household-level data?
- **Firm outcomes:** do leverage, investment, productivity, and export behavior organize into stable subnetworks?

Interpretation

These questions are often more informative than asking whether one specific edge is present.

Useful summaries of an estimated graph

Once we have learned a graph $\widehat{G}$, we may want to summarize it.

Local summaries

- degree: how many neighbors a node has,
- weighted degree / strength,
- local clustering coefficient,
- neighborhood of a given variable.

Global summaries

- connected components,
- communities or clusters,
- graph density,
- distances and diameter.

R

- `igraph`: network summaries, communities
- `qgraph`: visualization

Python

- `networkx`: standard graph analysis
- `leidenalg`: community detection

A better way to report edges

In Gaussian graphical models, it is often more informative to report **partial correlations** and not only the binary graph.

Estimated partial correlation

If $\widehat{K}$ is the estimated precision matrix, define

$$\widehat{\rho}_{ij|V\setminus\{i,j\}} = -\frac{\widehat{K}_{ij}}{\sqrt{\widehat{K}_{ii}\widehat{K}_{jj}}} \in [-1, 1].$$

The partial correlation tells us the **sign** and **magnitude** of the fitted conditional association.

For interpretation it is often useful to visualize both: the sparse graph structure and the estimated edge weights.

Example of graph analysis in R

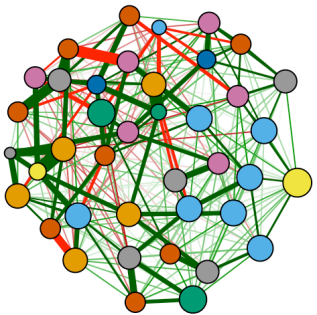
For network analysis we can convert the estimated graph to an igraph object.

```
library(huge)
library(qgraph)
library(igraph)

data(stockdata)
x <- scale(stockdata$data[,1:40])

out <- huge(x, method = "glasso")
Khat <- out$icov[[10]]
Khat <- (Khat + t(Khat))/2

Gq <- qgraph(-cov2cor(Khat), layout = "spring")

W <- getWmat(Gq)
G <- graph_from_adjacency_matrix(W != 0, mode = "undirected", diag = FALSE)

edge_density(G, loops = FALSE)
igraph::degree(G)
components(G)$csize
```

Centrality analysis

The degree is one simple measure of centrality.

```
deg <- igraph::degree(G, mode = "all")
which(deg == max(deg))

plot(G,
     rescale=FALSE,
     vertex.size = 5*deg,
     vertex.color = deg,
     vertex.label = NA,
     layout = 5*Gq$layout)
```



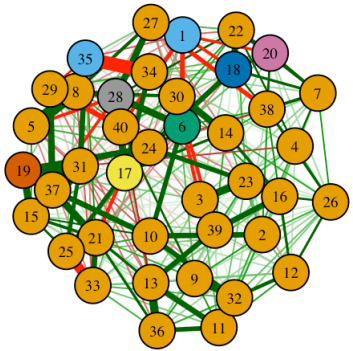
A variable with high degree is conditionally linked to many others. In an economic application, such variables may act as useful summary nodes of the dependence structure.

There are many other measures of centrality, which we do not discuss here in detail.

Communities

Many estimated networks have modules that are densely connected internally but weakly connected to the rest.

```
lay <- as.matrix(Gq$layout)

plot(G,
     vertex.color = cut_at(ceb, 10),
     vertex.label.cex = .6,
     layout = lay)
```



Interpretation

In applications, communities may correspond to sectors, groups of related macro indicators, or other latent blocks of variables with similar dependence patterns.

Take-away

- Undirected graphs encode conditional independence statements.
- In Gaussian graphical models, missing edges correspond to zeros in the precision matrix.
- The Hammersley–Clifford theorem links graph separation with factorization.
- In high dimensions, sparse estimation requires regularization such as graphical lasso.
- Non-paranormal methods make the same ideas more robust for heavy-tailed and non-Gaussian data.

Main lesson of today: conditional independence becomes a network, and sparsity makes that network learnable.

Looking ahead

Next lecture

We move to directed graphs. Instead of symmetric conditional dependence, we will study asymmetry, causal ordering, and interventions.

Coming next

- DAGs and d-separation,
- Markov equivalence,
- structural equation models,
- soft interventions and policy interpretation.