STA 437/2005: Methods for Multivariate Data

Week 11-12: Conditional independence and graphical models

Piotr Zwiernik

University of Toronto

Conditional independence

Basic definitions

Random vector and independence

Let (X, Y) be a vector of two random variables.

Joint distribution

Density function $f_{XY}(x, y)$ if continuous.

Probability mass function $f_{XY}(x, y) = \mathbb{P}(X = x, Y = y)$ if discrete.

Marginal distribution

continuous: $f_X(x) = \int_{\mathbb{R}} f_{XY}(x, y)\mathrm{d}y$.

discrete: $f_X(x) = \sum_y f_{XY}(x, y) = \mathbb{P}(X = x)$.

This can be generalized to random vectors.

Independence

If $f_{XY}(x, y)$ is the joint density (or PMF) of (X, Y) then X and Y are independent if and only if

$$f_{XY}(x, y) = f_X(x) f_Y(y) \qquad \text{for all } x, y.$$

We write $X \perp\!\!\!\perp Y$.

Recall:

$$\text{cov}(X, Y) = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y) \quad \text{and} \quad \mathrm{var}(X) = \text{cov}(X, X).$$

The correlation $\rho_{X,Y}$ between X, Y is:

$$\rho_{X,Y} \;=\; \frac{\text{cov}(X,Y)}{\sqrt{\mathrm{var}(X)\mathrm{var}(Y)}} \;\in\; [-1, 1].$$

If $X \perp\!\!\!\perp Y$ then $\rho_{X,Y} = 0$.

(but in general not the other way around, see slide 13)

Conditional distribution

Conditional distribution

In the discrete case the conditional probability mass function is defined as

$$f_{X|Y}(x|y) = \mathbb{P}(X = x|Y = y) = \frac{\mathbb{P}(X=x, Y=y)}{\mathbb{P}(Y=y)}$$

for all x, y such that $\mathbb{P}(Y = y) > 0$ and so

$$f_{X|Y}(x|y) = \frac{f_{XY}(x,y)}{f_Y(y)} \text{ for all } x, y \text{ s.t. } f_Y(y) > 0.$$

In the continuous case we use the same definition.

Important reformulation of independence

$X \perp\!\!\!\perp Y$ if and only if $f_{X|Y}(x|y) = f_X(x)$.
(knowing Y brings no extra information about X)

A cautionary note

Note: Large $f_{X|Y}(x|y)$ does not imply large $f_{Y|X}(y|x)$.

Example: A medical test for a disease D has outcomes $+$ and - with probabilities

	D	D^c
$+$	.009	.099
-	.001	.891

As needed $\mathbb{P}(+|D) = 0.9$ and $\mathbb{P}(-|D^c) = 0.9$. However, $\mathbb{P}(D|+) \approx 0.08$ (!)

Conditional independence

X, Y, Z random variables.

X is independent of Y given Z (write $X \perp\!\!\!\perp Y|Z$) if

$$f_{XY|Z}(x, y|z) \; = \; f_{X|Z}(x|z)f_{Y|Z}(y|z) \qquad \text{for every } z.$$

Important reformulation of independence

$X \perp\!\!\!\perp Y|Z$ if and only if $f_{X|Y,Z}(x|y, z) = f_{X|Z}(x|z)$.
(if we observed Z, extra information about Y brings no extra information about X)

Testing independence

Recall: A statistical test

Given a statistical hypothesis $H_0 : \theta \in \Theta_0$, $H_1 : \theta \in \Theta_1$, a statistical test consists of a test statistics $T(X^{(1)}, \dots, X^{(n)})$ and a rejection region, typically of the form

$$R = \{T(X^{(1)}, \dots, X^{(n)}) > t\}.$$

If the null hypothesis is true T is unlikely to take large values.

Type I error: $\mathbb{P}(T \in R | H_0)$

although H_0 is true, it is rejected

Type II error: $\mathbb{P}(T \notin R | H_1)$

although H_0 is false, it is retained

A good test should minimize probabilities of both types of errors.

Testing independence

Data: $(X_1, Y_1), \ldots, (X_n, Y_n) \overset{iid}{\sim} P_{X,Y}$.

Goal: Decide whether $X \perp\!\!\!\perp Y$.

Statistical test: $\quad H_0 : X \perp\!\!\!\perp Y, \quad H_A : X \not\perp\!\!\!\perp Y$

There are many tests of independence.

We discuss some examples.

Test for vanishing correlation

Fisher's z-transform test for Gaussian data

Let r_n is the sample correlation coefficient from an *iid* sample $(X^{(i)}, Y^{(i)})$.

Define $Z_n = \frac{1}{2}\log\left(\frac{1+r_n}{1-r_n}\right)$.

If (X, Y) is bivariate normal with correlation ρ then Z_n has asymptotically normal distribution with mean $\frac{1}{2}\log\left(\frac{1+\rho}{1-\rho}\right)$ and variance $\frac{1}{n-3}$.

Fisher's z-transform test is implemented in R as `cor.test`.

Non-gaussianity may invalidate the test and affect its power.

Basic nonparametric test

Kendall's tau test for non-Gaussian data

Suppose a bivariate sample (x_i, y_i) for $i = 1, \dots, n$ is given.

Pair (x_i, y_i) , (x_j, y_j) is concordant if $(x_i, y_i) < (x_j, y_j)$ or $(x_i, y_i) > (x_j, y_j)$. Otherwise discordant.

Define $\tau_{XY} = \frac{(\#\text{concordant}) - (\#\text{discordant})}{\binom{n}{2}} \in [-1, 1]$.

Test based on Kendell's τ statistic is implemented in R as `cor.test`.

```
> set.seed(1); n <- 200; rho <- 0.2; Z <- runif(n);
> X <- runif(n)^2+sqrt(rho)*Z; Y <- runif(n)+sqrt(rho)*Z
> cor.test(X, Y, method = "pearson")$p.value
[1] 0.03417231
> cor.test(X, Y, method = "kendall")$p.value
[1] 0.01100592
```

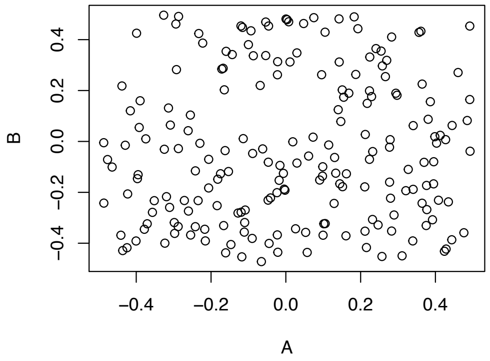
τ_{XY} is asymptotically normal under independence, which is used to construct p-values.

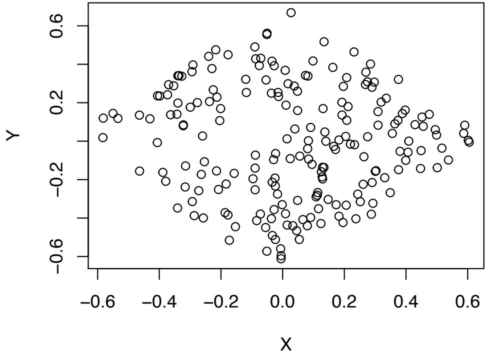
Non-Gaussianity issue

Vanishing covariance does not imply independence!

```
# generate sample from two uncorrelated but dependent random variables
> set.seed(1); n <- 200
> A <- runif(n)-1/2; B <- runif(n)-1/2
> X <- t(c(cos(pi/4),-sin(pi/4)) %*% rbind(A,B))
> Y <- t(c(sin(pi/4),cos(pi/4)) %*% rbind(A,B))
> cor.test(X,Y, method = "pearson")
# Pearson's product-moment correlation
data:  X and Y
t = -0.84711, df = 198, p-value = 0.398
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1971897  0.0793095
sample estimates:
        cor
-0.06009275
```

X and Y are uncorrelated but dependent!





We see that X and Y are highly dependent.

Test based on distance correlation (very high level)

Distance correlation $\mathcal{R}(X, Y)$ provides a test which applies when X, Y are two random **vectors** of any dimensions.

$\mathcal{R}(X, Y) = 0$ if and only if X and Y are independent.

The sample version of $\mathcal{R}(X, Y)$ gives a nonparametric test of independence.

```
> library(energy); set.seed(1); n <- 200
> A <- runif(n)-1/2; B <- runif(n)-1/2
> X <- t(c(cos(pi/4),-sin(pi/4)) %*% rbind(A,B))
> Y <- t(c(sin(pi/4),cos(pi/4)) %*% rbind(A,B))
> dcor.test(X,Y,R=1000)

# dCor independence test (permutation test)
data:  index 1, replicates 1000
dCor = 0.21161, p-value = 0.004995
sample estimates:
      dCov        dCor     dVar(X)     dVar(Y)
0.03999654 0.21160982 0.17870935 0.19990601
```

Tends to be more powerful than the test based on Kendall's tau.

Another cautionary example

Bowman&Azzalini (1997) analyse aircraft wing span and speed data.

```
> library(sm); set.seed(1);
> X <- aircraft$Span
> Y <- aircraft$Speed
> cor.test(X,Y)$p.value
[1] 0.7816014
> dcor.test(X,Y,R=1000)$p.value
[1] 0.000999001
```

Appendix: Asymptotic Likelihood Ratio Test

Consider a parametric model $\{\mathbb{P}_\theta : \theta \in \Theta\}$ and let $\Theta_0 \subseteq \Theta$. Test $H_0 : \theta^* \in \Theta_0$.

Let $\ell(\theta) = \sum_{i=1}^n \log f(\mathbf{x}_i; \theta)$ be the log-likelihood. Compute the statistics:

$$\lambda_n := 2(\sup_{\theta\in\Theta} \ell(\theta) - \sup_{\theta\in\Theta_0} \ell(\theta))$$

Wilks' theorem (simplified version)

Under H_0, if Θ and Θ_0 are regular $\lambda_n \xrightarrow{d} \chi^2_d$, where d $= \dim(\Theta) - \dim(\Theta_0)$ is the difference in the number of parameters.

Testing independence for discrete distributions

G-test

In the context of count data the LRT is called the G-test.

Example (Testing Independence):

Suppose we have X with k possible values and Y with l possible values.

The saturated model for (X, Y) has dimension $kl - 1$.

The independence model $X \perp\!\!\!\perp Y$ has dimension $(k-1) + (l-1)$.

We could use LRT with $d = (k-1)(l-1)$.

Tests for discrete data

χ^2-test for discrete data (uses `library(DescTools)`)

```
> M <- as.table(rbind(c(762, 327, 468), c(484, 239, 477)))
> dimnames(M) <- list(gender = c("F", "M"),
+                     party = c("Democrat","Independent", "Republican"))

# Perform the LRT (G-test)
> GTest_result <- GTest(M)
# Print results
> print(GTest_result)

Log-likelihood ratio (G-test) test of independence without correction

data:  M
G = 30.017, X-squared df = 2, p-value = 3.034e-07
```

`df = 2` is the difference between 5 (saturated model) and 3 (independence)

Testing conditional independence

Testing conditional independence

Testing conditional independence is hard in general.

(For discrete data we use the LRT.)

Some parametric tests are implemented in the library `bnlearn`.
Many non-parametric methods have been implemented in `CondIndTest`

```
> library(CondIndTests); library(bnlearn); set.seed(1); n <- 100
> Z <- rnorm(n); X <- 4 + 2 * Z + rnorm(n); Y <- 3 * X^2 + Z + rnorm(n)
> CondIndTest(X,Y,Z, method = "KCI")$pvalue
[1] 2.419926e-10
> bnlearn::ci.test(X,Y,Z)$p.value
[1] 1.15458e-25
```

See Section 3 in: C. Heinze-Deml, J. Peters, N. Meinshausen, Invariant Causal Prediction for Nonlinear Models, Journal of Causal Inference, 2018.

See: `http://www.bnlearn.com/documentation/man/conditional.independence.tests.html`

Simpson's paradox: UC Berkeley admissions example

The admission figures of the grad school at UC Berkeley in 1973: 8442 (44%) men, 4321 (35%) women admitted.

The same data conditioned on the department are:

Department	Men		Women	
	Applicants	Admitted	Applicants	Admitted
A	825	62%	108	**82%**
B	560	63%	25	**68%**
C	325	**37%**	593	34%
D	417	33%	375	**35%**
E	191	**28%**	393	24%
F	373	6%	341	**7%**

"Measuring bias is harder than is usually assumed, and the evidence is sometimes contrary to expectation."

(Bickel et al, *Sex Bias in Graduate Admissions: Data From Berkeley*, Science, 1975)

In R:

```
> library(gRim); data(UCBAdmissions)
> gRim::ciTest(as.data.frame(UCBAdmissions),set=~Gender+Admit+Dept)

set: [1] "Gender" "Admit"  "Dept"
Testing Gender _|_ Admit | Dept
Statistic (DEV):    0.000 df: 6 p-value: 1.0000 method: CHISQ

Slice information:
  statistic p.value df Dept
1         0       1  1    A
2         0       1  1    B
3         0       1  1    C
4         0       1  1    D
5         0       1  1    E
6         0       1  1    F
```

In the last part a simple independence test is performed for each department separately.

Conditional independence for Gaussian distributions

Recall: Marginal and conditional distributions

Split X into two blocks $X = (X_A, X_B)$. Denote

$$\mu = (\mu_A, \mu_B) \qquad \text{and} \qquad \Sigma = \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix}.$$

Marginal distribution

$X_A \sim N_{|A|}(\mu_A, \Sigma_{AA})$

Conditional distribution

$X_A | X_B = x_B \sim N_{|A|}\left(\mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(x_B - \mu_B), \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}\right)$

- Note that the conditional covariance is constant.

Conditional independence

Independence and conditional independence

$X_i \perp\!\!\!\perp X_j$ if and only if $\Sigma_{ij} = 0$.

$X_i \perp\!\!\!\perp X_j | X_C$ if and only if $\Sigma_{ij} - \Sigma_{i,C}\Sigma_{C,C}^{-1}\Sigma_{C,j} = 0$

Let $R = V \setminus \{i,j\}$. The following are equivalent:

- $X_i \perp\!\!\!\perp X_j | X_R$
- $\Sigma_{ij} - \Sigma_{i,R}\Sigma_{R,R}^{-1}\Sigma_{R,j} = 0$
- $(\Sigma^{-1})_{ij} = 0$

Useful: `https://en.wikipedia.org/wiki/Block_matrix#Block_matrix_inversion`

Undirected graphical models

Graph factorizations

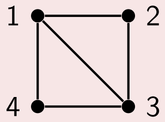
Factorization

G = an undirected graph with nodes $\{1, \ldots, m\}$ and cliques $C_1, \ldots, C_k$.
We say that density $f(\boldsymbol{x})$ **factorizes according to G** if for all $\boldsymbol{x} \in \mathcal{X}$

$$f(\boldsymbol{x}) \;=\; \phi_{C_1}(\boldsymbol{x}_{C_1}) \cdots \phi_{C_k}(\boldsymbol{x}_{C_k}),$$

where $\phi_C(\boldsymbol{x}_C) \geq 0$. (a notion of simplicity)

For example



$$f(\boldsymbol{x}) = \phi_{123}(x_1, x_2, x_3)\phi_{134}(x_1, x_3, x_4).$$

Exercise: This gives an alternative characterisation of $X_2 \perp\!\!\!\perp X_4 | (X_1, X_3)$.

Hammersley-Clifford theorem

Let $f > 0$ be a dentity function for $\boldsymbol{X} = (X_1, \ldots, X_m)$. Then the following are equivalent:

(F) f factorizes according to G.

(G) $X_A \perp\!\!\!\perp X_B | X_C$ whenever C separates A and B in G.

(P) $X_i \perp\!\!\!\perp X_j | X_{V \setminus \{i,j\}}$ whenever i, j not connected by an edge in G.

The graph represents conditional independence.

Graphical model $\mathcal{M}(G)$

G a graph with m nodes representing a random vector $X = (X_1, \dots, X_m)$.

Definition (Graphical model $\mathcal{M}(G)$)

$\mathcal{M}(G)$ is the family of all distributions of X that factorize according to G.

By the Hammersley-Clifford theorem this can be equivalently described by conditional independences. (we work with positive distributions only)

Model families that admit suitable factorizations are described in later parts:

1. log-linear models for multivariate discrete data
2. graphical Gaussian models for multivariate Gaussian data.

This drives modelling for more complicated (non-parametric) settings.

Gaussian graphical models (GGMs)

The Gaussian case

For a Gaussian distribution in $\mathcal{M}(G)$:

The non-edges of G correspond to conditional independences $X_i \perp\!\!\!\perp X_j | X_{V\setminus\{i,j\}}$ or equivalently $\mathbf{K_{ij} = 0}$.

Exercise: How $K_{ij} = 0$ implies that the density factorizes into factors $V \setminus \{i\}$, $V \setminus \{j\}$.

Two main estimation problems:

Consider an *iid* sample $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from a distribution in $\mathcal{M}(G)$.

(i) Estimate Σ for a fixed graph G.

(ii) Learn the underlyin graph if it is unknown.

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)

The Gaussian likelihood function

The sample covariance matrix of the sample $\mathbf{x}_1, \dots, \mathbf{x}_n$ is

$$S = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T.$$

Recall that the log-likelihood is

$$\log L(\mu, K) \;=\; \tfrac{n}{2}\log\det K - \tfrac{n}{2}\mathrm{trace}(KS) - \tfrac{n}{2}(\bar{\mathbf{x}} - \mu)^T K(\bar{\mathbf{x}} - \mu).$$

For fixed K we get $\hat{\mu} = \bar{\mathbf{x}}$ giving the profile likelihood

$$\log L(\hat{\mu}, K) \;=\; \tfrac{n}{2}\log\det K - \tfrac{n}{2}\mathrm{trace}(KS).$$

Maximizing the likelihood over $\mathcal{M}(G)$

$\mathcal{M}(G)$ consists of PD matrices K such that $K_{ij} = 0$ for $ij \notin E$.

Optimizing $\log L(\hat{\mu}, K)$ over $\mathcal{M}(G)$ is a convex optimization problem.

The MLE is the unique point $\hat{\Sigma}$ (with $\widehat{K} = \widehat{\Sigma}^{-1}$) such that:

(i) $\hat{\Sigma}_{ii} = S_{ii}$ for all i, and $\hat{\Sigma}_{ij} = S_{ij}$ for all edges $ij \in G$,

(ii) $\hat{K}_{ij} = 0$ for all $ij \notin G$.

Maximization

Typically numerically using a block coordinate-descent scheme (`ggmfit`).

The deviance

Gives the likelihood ratio statistic to test against the unconstrained model.

Example: Financial market data

Using gRbase, gRim, RBGL:

```
> data(EuStockMarkets)  # European stock market index data
> df <- as.data.frame(EuStockMarkets)  # Convert to a dataframe
> df <- scale(df)  # Standardize data
# Define graphical model structure (assumed dependencies)
> glist <- list(c("DAX", "SMI", "CAC"), c("SMI", "CAC", "FTSE"))
# Fit Gaussian graphical model
> gen.fit <- cmod(glist, data=df, fit=TRUE)
# Deviance and goodness-of-fit test
> gen.fit$fitinfo$dev; 1 - pchisq(gen.fit$fitinfo$dev, df=1)
[1] 0.6793619
[1] 0.4098065
# Visualize estimated partial correlation network
> S <- solve((gen.fit$fitinfo$K + t(gen.fit$fitinfo$K))/2)
> qgraph::qgraph(S, graph = "pcor")
```

Structure learning in GGMs

Model selection methods

How to learn the graph:

- Stepwise methods,
- Convex optimization,

Appendix: Information criteria in model selection

Information criteria provide a popular method for model selection.

Consider a selection of models $M_1, \dots, M_r$ with dimensions $d_1, \dots, d_r$.

Given a sample $\mathbf{x}_1, \dots, \mathbf{x}_n$, we compute the MLE for each of the models; $\ell(M_i)$ denotes the value of the log-likelihood.

Then

$$\begin{aligned} AIC_i &= -2\ell(M_i) + 2d_i \\ BIC_i &= -2\ell(M_i) + \log(n)d_i \end{aligned}$$

We pick the model with the lowest Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC).

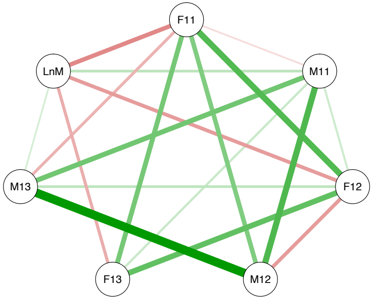
Dimension of a Gaussian Graphical Model

For GGMs the dimension is simply $|V| + |E|$.

Stepwise methods

The stepwise function in gRim performs stepwise model selection based on a variety of criteria (AIC, BIC, etc)

```
sat.carc <- cmod(~.^.,data=carcass); n <- nrow(carcass)
test.carc <- stepwise(sat.carc,details=1,criterion="aic",type="decomposable",k
    =log(n))
# plot(test.carc) or using the qgraph package
Sigma.hat <- solve(test.carc$fitinfo$K)
qgraph::qgraph(Sigma.hat,graph="pcor")
```



Learning the graph in high-dimension

Graphical LASSO

If the dimension m is large then the number of possible models is too large.

Instead, following the same idea as in the LASSO regression we maximize

$$L_{\text{pen}}(K, \hat{\mu}) \; = \; \log\det(K) - \text{trace}(SK) - \lambda\|K\|_1,$$

where $\|K\|_1 = \sum_{i \neq j} |K_{ij}|$ and λ is a fixed penalty parameter.

Implementation in R

See the packages glasso and EBICglasso (finds an optimal λ).

```
> S <- cor(data)
> lambda_opt <- 0.1  # Set a small regularization parameter
> glasso_fit <- glasso(S, rho=lambda_opt)
```

Log-linear models

Examples of log-linear models

Random vector $X = (X_1, \dots, X_m)$ with state space $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_m, \quad |\mathcal{X}_i| < +\infty$.

Generators: Denote by $\mathcal{C}$ a set of subsets of $\{1, \dots, m\}$.

The model: $p(x) \;=\; \prod_{C \in \mathcal{C}} \phi_C(x_C), \quad$ where $\phi_C > 0$.

Main effect model: $\mathcal{C} = \{\{1\}, \dots, \{m\}\}$.

Pairwise interaction model: $\mathcal{C} = \{$all pairs$\}$.

- more generally: edges of a graph with m nodes.

Graphical models: $\mathcal{C}$ be the set of maximal cliques of G.

Fitting log-linear models in R

The function `loglin()` fits general log-linear models. We focus here on the ones represented by graphs, that is, graphical or pairwise.

The `gRim` package has a function `dmod()` to define and fit hierarchical log-linear models for a fixed set of generators.

```
> data(lizard)
> mliz <- dmod(list(c("species","height"),c("species","diam")),data=lizard)
> mliz

Model: A dModel with 3 variables
 -2logL    :         1604.43 mdim :    5 aic :       1614.43
 ideviance :           23.01 idf  :    2 bic :       1634.49
 deviance  :            2.03 df   :    2
```

On `data(lizard)`: Behaviour characteristics of 409 lizards were recorded: species (`anoli`, `dist`), perch diameter (<=4, >4), and perch height (`>4.75`,`<=4.75`).

Testing conditional independence

If the data is given in form of a contingency table it is convinient to use gRim's `ciTest_table`.

```
> ciTest_table(lizard,set=c("height","diam","species"))
Testing height _|_ diam | species
Statistic (DEV):    2.026 df: 2 p-value: 0.3632 method: CHISQ
Slice information:
  statistic p.value df species
1     0.178  0.6731  1   anoli
2     1.848  0.1741  1    dist


> ciTest_table(lizard,set=c("diam","species","height"))
Testing diam _|_ species | height
Statistic (DEV):   14.024 df: 2 p-value: 0.0009 method: CHISQ
Slice information:
  statistic   p.value df height
1     2.903 0.0884377  1  >4.75
2    11.122 0.0008533  1 <=4.75
```

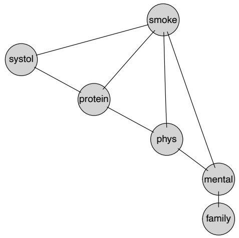
Notice the `df` calculation, which in general may be complicated.

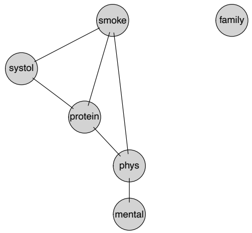
Stepwise methods for discrete graphical models

For stepwise methods typically the penalized likelihood criteria like BIC or AIC are used.

Stepwise methods either start with the full graph or the empty graph.

```
> data(reinis); m.init <- dmod(~.^., data=reinis)
> m.reinis <- stepwise(m.init) # AIC criterion
> m.reinis.2 <- stepwise(m.init,k=log(sum(reinis))) # BIC criterion
> plot(m.reinis); plot(m.reinis.2)
```





Modelling large data sets

With >10 variables stepwise methods are computationally prohibitive.

Possible strategies:

1. Use MCMC (`BDgraph` package)
2. Focus on decomposable models (`gRapHD` package).
3. Rely on simpler models (e.g. pairwise interaction models).
4. Restrict to very simple graphs (e.g. trees).

This is an active research area.

Binary Ising model

Consider a log-linear model with only pairwise interactions.

If $\mathcal{X} = \{-1, 1\}^m$ then

$$f(x) \; = \; \tfrac{1}{Z(h,B)} \exp\left(\sum_i h_i x_i + \sum_{i<j} \beta_{ij} x_i x_j\right),$$

where $h \in \mathbb{R}^m$ and $B = [\beta_{ij}]$ has zeros on the diagonal.

Likelihood is hard to handle because $Z(h, B)$ is intractable if m is large.

This motivates pseudo-likelihood methods.

Pseudo-likelihood approach

Logistic regression

Denoting $\eta_k = h_k + \sum_{i \neq k} \beta_{ik} x_i$, the full conditional distributions satisfy

$$\log p(x_k | x_{-k}) = \eta_k x_k - \log(e^{-\eta_k} + e^{\eta_k})$$

and so, if $p = p(1|x_{-k})$, then $\log \frac{p}{1-p} \;=\; 2\eta_k \;=\; 2h_k + \sum_{i \neq k} 2\beta_{ik} x_i.$

ℓ_1-regularized logistic regression (Ravikumar et al, 2010)

Having observed $\mathbf{x}_1, \ldots, \mathbf{x}_n$, to learn the support of B, we can optimize for each k the logistic regression given the data. Alternatively we can maximize

$$\sum_{i=1}^{n} \sum_{k \in V} \log p(x_k^{(i)} | x_{-k}^{(i)}) \;-\; \lambda \|B\|_1.$$

Using IsingFit

```
> install.packages("IsingFit"); library(IsingFit)
> ncsdata=read.table(file="./data/DepressionAnxiety.txt")
> colnames(ncsdata)=c("depr", "inte", "weig", "mSle", "moto", "mFat", "repr",
    "conc", "suic", "anxi", "even", "ctrl", "edge", "gFat", "irri", "gCon", "
    musc", "gSle") #Define variable names.
# remove two variables that are perfectly correlated with each other in the
    sample
> X <- (ncsdata[,-(10:11)])
# run the high-dimensional Ising model selection problem
> Res <- IsingFit(X,gamma = 0.5, plot=FALSE)
# compare with the correlation network
> lay <-averageLayout(Res$weiadj,cor(X), layout = "spring", repulsion = 1)
> qgraph(cor(X),layout=lay,labels=colnames(X))
> qgraph(Res$weiadj, layout = lay)
# both graphs appear on the next slide
```

For a discussion of this dataset see:

Borsboom and Cramer, Network analysis: an integrative approach to the structure of psychopathology, Annual review of clinical psychology, 9 (2013).

Two psychological disorders

About the study:

National Comorbidity Survey Replication (NCS-R data)

9282 observations of 18 binary variables such as:
`depr` (Depressed mood), `inte` (Loss of interest), etc

These are symptoms related to two disorders:
major depression and generalized anxiety disorder.

Bridge variables: sleep problems, fatigue, and concentration problems.

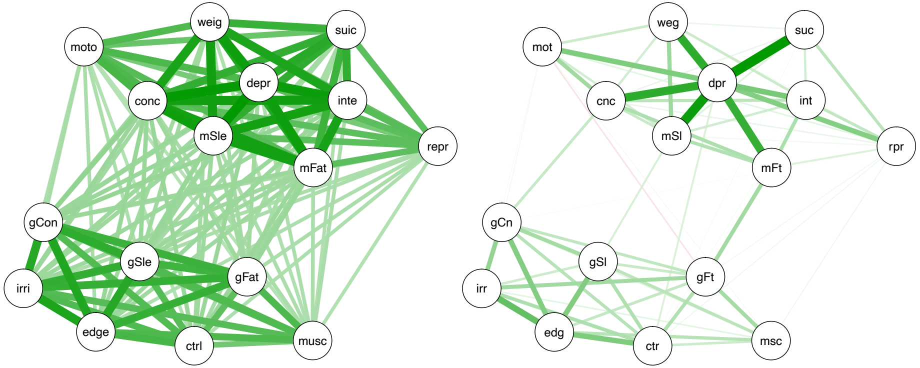
Two psychological disorders, continued

About the data:

Sparse contingency table: 872/65536 nonzero cells.

5667 out of 9282 respondents recorded no symptoms.

two variables perfectly correlated with each other and other seven variables.


weig
suic
moto
depr
conc
inte
mSle
repr
mFat
gCon
gSle
irri
gFat
edge
ctrl
musc
weg
suc
mot
dpr
cnc
int
mSl
rpr
mFt
gCn
gSl
irr
gFt
edg
ctr
msc

Gaussian copula graphical models

Nonparanormal distributions

$(X_1, \ldots, X_m)$ has nonparanormal distribution if $(f_1(X_1), \ldots, f_m(X_m))$ is Gaussian $N(\mu, \Sigma)$ for some monotone functions $f_1, \ldots, f_m$.

- This is the same as the Gaussian copula model; thus assume $\mu = 0$, $\Sigma_{ii} = 1$.

The functions f_i are treated as nuisance parameters. The correlation matrix is estimated directly using rank correlation statistics:

We can estimate the correlation matrix of $f(X)$ without knowing f!!

If $\tau_{ij} = \mathrm{cor}(\mathrm{sgn}(X_i - X_i'), \mathrm{sgn}(X_j - X_j'))$ is the Kendall's tau coefficient then

$$\Sigma_{ij} := \mathrm{cor}(f(X_i), f(X_j)) \;=\; \sin\left(\tfrac{\pi}{2}\tau_{ij}\right).$$

- τ_{ij}'s are invariant under monotone transformations on X_i's.

Note that graphical models for X and $f(X)$ are identical.

Estimating nonparanormal graphical models

Compute Kendall's tau coefficients from the data (c.f. Slide 10).

Define $\hat{S}^{\tau} = \sin(\frac{\pi}{2}\hat{\tau}_{ij})$.

Concentration analysis based on U-statistics shows that

$$\max_{ij} |\hat{S}^{\tau}_{ij} - \Sigma^0_{ij}| = \mathcal{O}\left(\sqrt{\frac{\log(mn)}{n}}\right).$$

Based on $\hat{S}^{\tau}$ we can now employ any standard Gaussian procedure to learn the graph, e.g., graphical lasso.

Application using huge package

Stock market data: closing prices from all stocks in the S&P 500 for all the days that the market was open between Jan 1, 2003 and Jan 1, 2008.

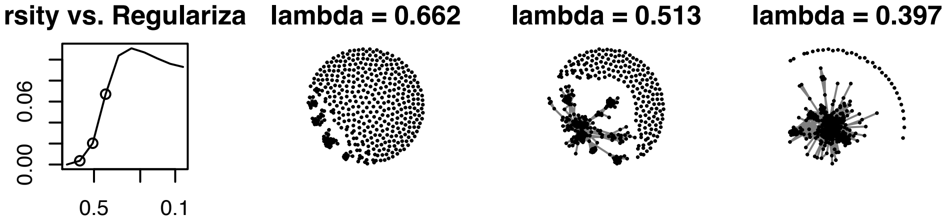
```
> library(huge); data(stockdata) # Load the data
> x = log(stockdata$data[2:1258,]/stockdata$data[1:1257,]) # Preprocessing
> colnames(x) <- stockdata$info[,1]
> x.npn = huge.npn(x, npn.func="skeptic") # Nonparanormal + Kendal's tau
> out.npn = huge(x.npn,method = "glasso")
> plot(out.npn)
# plot the graph corresponding to lambda=0.397
> Khat <- (out.npn$icov[[15]]+t(out.npn$icov[[15]]))/2; colnames(Khat) <-
    rownames(Khat) <- colnames(x)
> dev.off(); qgraph::qgraph(-cov2cor(Khat),layout="spring")
```

This is a rather large example to be handled by qgraph directly.

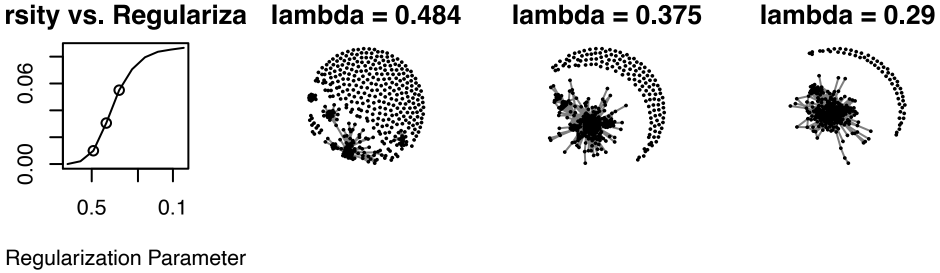
```
> qgraph(x.npn,graph="glasso",sampleSize=nrow(x),layout="spring")
```

Zhao, T., Liu, H., Roeder, K., Lafferty, J., & Wasserman, L. (2012). The huge package for high-dimensional undirected graph estimation in R. Journal of Machine Learning Research.

The non-paranormal approach gives:



For comparison a direct Gaussian approach gives:


rsity vs. Regulariza
Regularization Parameter
lambda = 0.484
lambda = 0.375
lambda = 0.29