

# *Statistical Learning for Data Science*

*MSDS534*

## Lecture 06: Graphical Models

Department of Statistics & Biostatistics  
Rutgers University

# Acknowledgement

- Some of the figures in this presentation are taken from *Elements of Statistical Learning* (Springer, 2009) with permission from the authors: T. Hastie, R. Tibshirani and J. Friedman.
- Some of the figures in this presentation are taken from *Pattern Recognition and Machine Learning* (Springer, 2006) with permission from the author: Christopher M. Bishop.

# Reading Assignments

- ESL: §17.1–17.3
- PRML: §8.1–8.3

# Graphical Models

- They provide a simple way to visualize the structure of a probabilistic model and can be used to design and motivate new models.
- Insights into the properties of the model, including conditional independence properties, can be obtained by inspection of the graph.
- Complex computations, required to perform inference and learning in sophisticated models, can be expressed in terms of graphical manipulations, in which underlying mathematical expressions are carried along implicitly.

# Graphs and Graphical Models

- A graph comprises **nodes** (also called **vertices**) connected by edges (also known as **links** or **arcs**).
- In a probabilistic graphical model, each node represents a random variable, and the edges express probabilistic relationships between these variables.
- **Bayesian networks**, also known as **directed graphical models**: the edges of the graphs have a particular directionality indicated by arrows.
- **Markov random fields**, also known as **undirected graphical models**: the edges do not carry arrows and have no directional significance.
- Directed graphs are useful for expressing causal relationships between random variables. **Be careful!**
- Will focus on continuous, and in particular, Gaussian graphical models. However, to better understand the conditional probabilities calculation, it might be easier to have discrete random variables in mind.

- 1 **Bayesian Networks/Directed Graphical Models**
- 2 Conditional Independence
- 3 Markov Random Fields/Undirected Graphical Models

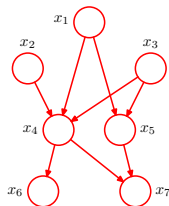
# Bayesian Networks/Directed Graphical Models

- Suppose there are  $K$  random variables  $\{x_1, \dots, x_K\}$ .
- The complete factorization of the joint density  $p(x_1, \dots, x_K)$  is given by

$$p(x_1, \dots, x_K) = p(x_K|x_1, \dots, x_{K-1}) \cdots p(x_2|x_1)p(x_1).$$

- Can use a directed graph to introduce additional structure into and to simplify this factorization.
- According to the graph, the joint density  $p(x_1, \dots, x_7)$  is factorized as

$$p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5).$$



- If there is a edge going from a node  $a$  to a node  $b$ , then we say that node  $a$  is the **parent** of node  $b$ , and we say that node  $b$  is the **child** of node  $a$ .
  - If there is a edge (arrow) going from a node  $a$  to a node  $b$ , we also say that  $a$  is at the **tail** of the edge, and  $b$  at the **head** of the edge.

# Bayesian Networks

- Order the nodes such that if  $b$  is a child of  $a$ , then  $b$  is ordered after  $a$ , i.e.  $b$  receives a larger ordering number.
- Assume there is **NO** directed cycles, i.e. there are no closed paths within the graph such that we can move from node to node along edges following the direction of the arrows and end up back at the starting node. Such a graph is also called a **directed acyclic graph**, or a **DAG**.
- The ordering is always possible for a DAG.
- Denote by  $\text{pa}_k$  the parents of the node  $x_k$ . The joint distribution is factorized as

$$p(\mathbf{x}) = \prod_{k=1}^K p(x_k | \text{pa}_k).$$

- The complete factorization corresponds to a complete DAG, with edges from every node to all the nodes after it.
- The factorization also leads to the **ancestral sampling** algorithm to sample from the joint distribution.
- Generally speaking, an undirected graphical model refers to the set of distributions satisfying the factorization induced by the graph.



# Gaussian DAG

- Suppose there is a DAG with  $K$  nodes. We consider the  $K$ -dimensional Gaussian distribution induced by the graph.
- Order the random variables  $x_1, \dots, x_K$  according to the ordering of the nodes.
- The conditional distribution of  $x_k$  given its parents  $\text{pa}_k$  is

$$x_k | \text{pa}_k \sim N \left( \sum_{j \in \text{pa}_k} w_{kj} x_j + b_k, v_k \right), \quad 1 \leq k \leq K.$$

- Equivalently,

$$x_k = \sum_{j \in \text{pa}_k} w_{kj} x_j + b_k + \sqrt{v_k} \epsilon_k,$$

where  $\epsilon_1, \dots, \epsilon_K$  are iid  $N(0, 1)$ .

# Build a Gaussian DAG

- If the graph is known, estimate the coefficients  $w_{kj}$  and the variance parameters  $v_k$  by LSE and RSS.
- If the graph is unknown, can use the following greedy algorithm.
  - 1 Suppose the current graph is  $\mathcal{G}^{(t)}$ .
  - 2 Try three possible operations on  $\mathcal{G}^{(t)}$ :
    - remove an edge,
    - add an edge if the graph is still acyclic,
    - reverse the direction of an edge if it is still acyclic.
  - 3 For each allowed operation in Step 2, calculate the BIC of the resulting graph. Select the one of the smallest BIC, call it  $\mathcal{G}^{(t+1)}$ .
  - 4 Iterate until none of the operations in Step 2 leads to a model with a smaller BIC.
- For a given graph  $\mathcal{G}$ , its BIC is calculated as

$$\text{BIC}(\mathcal{G}) = \sum_{k=1}^K \text{BIC}_k,$$

where

$$\text{BIC}_k = N \log \left[ \frac{1}{N} \sum_{i=1}^N \left( x_{ik} - \sum_{j \in \text{pa}_k} \hat{w}_{kj} x_j - \hat{b}_k \right)^2 \right] + |\text{pa}_k| \cdot \log N.$$

- 1 Bayesian Networks/Directed Graphical Models
- 2 Conditional Independence**
- 3 Markov Random Fields/Undirected Graphical Models

# Conditional Independence

- Two random variables  $a$  and  $b$  are independent if

$$p(a, b) = p(a)p(b) \quad \text{or} \quad p(a|b) = p(a).$$

- Now consider three random variables  $a$ ,  $b$  and  $c$ . The conditional density of  $(a, b)$  given  $c$  is

$$p(a, b|c) = \frac{p(a, b, c)}{p(c)}.$$

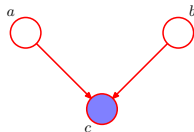
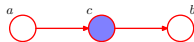
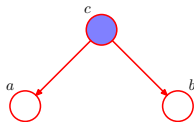
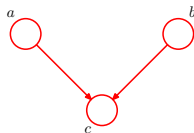
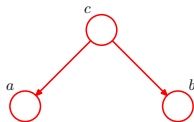
- We say  $a$  and  $b$  are conditional independent given  $c$  if

$$p(a, b|c) = p(a|c)p(b|c) \quad \text{or} \quad p(a|b, c) = p(a|c).$$

- Denote the conditional independence by  $a \perp\!\!\!\perp b \mid c$ .
- Denote the independence of  $a$  and  $b$  by  $a \perp\!\!\!\perp b \mid \emptyset$ , or simply  $a \perp\!\!\!\perp b$ .
- In general, it is often the case that  $a \not\perp\!\!\!\perp b$ , but we seek a  $c$  such that  $a \perp\!\!\!\perp b \mid c$ .

## Example: DAG with 3 Nodes

- Example 1.  $a \not\perp b$  and  $a \perp b | c$ . The node  $c$  is said to be **tail-to-tail** with respect to the path.
- Example 2.  $a \not\perp b$  and  $a \perp b | c$ . The node  $c$  is said to be **head-to-tail** with respect to the path.
- Example 3.  $a \perp b$ . Is  $a \perp b | c$ ? **NO!** The node  $c$  is said to be **head-to-head** with respect to the path.



## Example

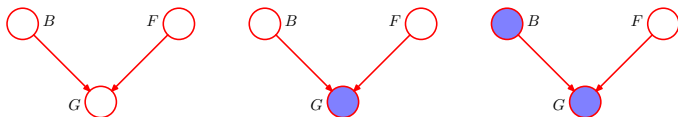
- Consider an example with three binary random variables relating to the fuel system on a car.
- The variable  $B$  represents the state of a battery that is either charged ( $B = 1$ ) or flat ( $B = 0$ ).
- The variable  $F$  represents the state of the fuel tank that is either full of fuel ( $F = 1$ ) or empty ( $F = 0$ ).
- The variable  $G$  represents the state of an electric fuel gauge and which indicates either full ( $G = 1$ ) or empty ( $G = 0$ ).
- Assume  $B$  and  $F$  are independent with  $P(B = 1) = P(F = 1) = 0.9$ .
- Assume that conditional on  $B$  and  $F$ ,

$$P(G = 1|B = 1, F = 1) = 0.8$$

$$P(G = 1|B = 1, F = 0) = 0.2$$

$$P(G = 1|B = 0, F = 1) = 0.2$$

$$P(G = 1|B = 0, F = 0) = 0.1$$



## Example

- Marginal and conditional probabilities of  $F = 0$ :

$$P(F = 0) = .1 \quad P(F = 0|G = 0) \approx .257 \quad P(F = 0|G = 0, B = 0) \approx .111$$

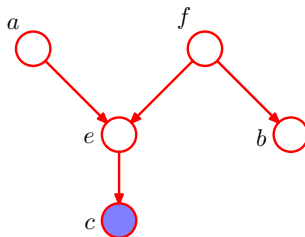
# D-Separation

- We say that node  $y$  is a **descendant** of node  $x$  if there is a path from  $x$  to  $y$  in which each step of the path follows the directions of the arrows.
- Consider a general directed graph in which  $A$ ,  $B$ , and  $C$  are arbitrary nonintersecting sets of nodes (whose union may be smaller than the complete set of nodes in the graph).
- Consider all possible paths from any node in  $A$  to any node in  $B$ . Any such path is said to be **blocked** if it includes a node such that either
  - (a) the arrows on the path meet either head-to-tail or tail-to-tail at the node, and the node is in the set  $C$ , or
  - (b) the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in the set  $C$ .
- If all paths are blocked, then  $A$  is said to be **d-separated** from  $B$  by  $C$ , and it codes that the joint distribution over all of the variables in the graph will satisfy  $A \perp\!\!\!\perp B \mid C$ .
- All the distributions that factorize according to the graph satisfy the conditional independence statements read from the graph.



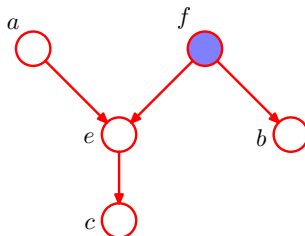
## Example: D-Separation

- Treat  $c$  as the conditioning set.
  - The path from  $a$  to  $b$  is not blocked by node  $f$  because it is a tail-to-tail node for this path, and it is not in the conditioning set.
  - The path from  $a$  to  $b$  is not blocked by node  $e$  because, although the latter is a head-to-head node, it has a descendant  $c$  in the conditioning set.
  - Thus the conditional independence statement  $a \perp\!\!\!\perp b \mid c$  does not follow from this graph, i.e.  $a \not\perp\!\!\!\perp b \mid c$ .



## Example: D-Separation

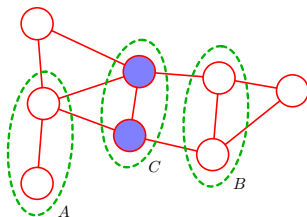
- Treat  $f$  as the conditioning set.
  - The path from  $a$  to  $b$  is blocked by node  $f$  because this is a tail-to-tail node, and is the conditioning set.
  - This path is also blocked by node  $e$  because  $e$  is a head-to-head node and neither it nor its descendant are in the conditioning set.
  - Thus the conditional independence statement  $a \perp\!\!\!\perp b \mid f$  follows from this graph.



- 1 Bayesian Networks/Directed Graphical Models
- 2 Conditional Independence
- 3 Markov Random Fields/Undirected Graphical Models**

# Markov Random Fields/Undirected Graphical Models

- A **Markov random field**, also known as a **Markov network** or an **undirected graphical model**, has a set of nodes each of which corresponds to a random variable, as well as a set of edges each of which connects a pair of nodes. The links are **undirected**, that is they do not carry arrows.
- It is convenient to begin with the conditional independence properties. (Compare with Bayesian networks!)
- Suppose that in an undirected graph we identify three sets of nodes, denoted  $A$ ,  $B$ , and  $C$ . Consider all possible paths that connect a node in set  $A$  to a node in set  $B$ . If all such paths pass through one or more nodes in set  $C$ , we say  $C$  **separates**  $A$  and  $B$ .
- Given an undirected graph  $\mathcal{G}$ , the Markov random field contains all distributions satisfying  $A \perp\!\!\!\perp B \mid C$ , as long as  $C$  separates  $A$  and  $B$ .



# Factorization

- A **clique** is a subset of the nodes in a graph such that there exists a edge between all pairs of nodes in the subset.
- A **maximal clique** is a clique such that it is not possible to include any other nodes from the graph in the set without it ceasing to be a clique.
- Denote a clique by  $C$  and the set of random variables in it by  $\mathbf{x}_C$ .
- The joint distribution is written as a product of **potential functions**  $\psi_C(\mathbf{x}_C)$  over the maximal cliques of the graph:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_C \psi_c(\mathbf{x}_c).$$

- Here the quantity  $Z$ , sometimes called the **partition function**, is a normalization constant and is given by

$$Z = \sum_{\mathbf{x}} \prod_C \psi_c(\mathbf{x}_c).$$

- **Hammersley-Clifford Theorem** asserts the equivalence between the conditional independence interpretation and the factorization, under the assumption that the joint densities/pmfs are positive everywhere.

# Pairwise Markov Independence

- Given an undirected graph  $\mathcal{G}$  with nodes  $V$ . Consider all the joint distributions of its nodes, such that  $a \perp\!\!\!\perp b \mid V \setminus \{a, b\}$  as long as  $a$  and  $b$  are not connected, i.e. no edge between  $a$  and  $b$ .
- This type of conditional independence is called the **pairwise Markov independence** or **pairwise Markov property** of  $\mathcal{G}$ .
- If  $a$  and  $b$  are not connected, then the set  $V \setminus \{a, b\}$  separates  $a$  and  $b$ .
- The aforementioned  $A \perp\!\!\!\perp B \mid C$  is called the **global Markov properties** of  $\mathcal{G}$ . It is obvious that the global Markov property implies the pairwise Markov property.
- The converse is also true, i.e. pairwise and global Markov properties are equivalent, when restricted to the densities/pmfs that are positive everywhere.

# Undirected Gaussian Graphical Model

- Suppose  $\mathbf{x} = (x_1, \dots, x_p)$  has a multivariate normal distribution  $N(\boldsymbol{\mu}, \Sigma)$ .
- The matrix  $\Theta = \Sigma^{-1}$  is called the **precision matrix**.
- $x_j \perp\!\!\!\perp x_k \mid \mathbf{x} \setminus \{x_j, x_k\}$  if and only if  $\theta_{jk} = 0$ . **HW problem.**
- Therefore, for an undirected Gaussian graphical model, the graph is characterized by the precision matrix.
- Let  $\mathcal{E}$  be the set of edges, i.e.  $(j, k) \in \mathcal{E}$  if  $\theta_{jk} \neq 0$ . Make the convention that all  $(j, j) \in \mathcal{E}$ .
- Given a sample  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ , let  $\mathbf{S}$  be the sample covariance matrix

$$\mathbf{S} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})'$$

- The log-likelihood of the precision matrix  $\Theta$  is given by

$$\ell(\Theta) = \log \det(\Theta) - \text{trace}(\mathbf{S}\Theta).$$

# Modified Regression Algorithm

- It can be verified that  $-\ell(\Theta)$  is convex in  $\Theta$ .
- If  $\mathcal{E}$  is known, the Lagrangian with respect to the given edge set  $\mathcal{E}$  is

$$L(\Theta, \Gamma) = \log \det(\Theta) - \text{trace}(\mathbf{S}\Theta) - \sum_{(j,k) \notin \mathcal{E}} \gamma_{jk} \theta_{jk}.$$

- The gradient condition is

$$\Theta^{-1} - \mathbf{S} - \Gamma = \mathbf{0},$$

where  $\Gamma$  is defined as:  $\Gamma[j, k] = \gamma_{jk}$  if  $(j, k) \notin \mathcal{E}$ , and  $\Gamma[j, k] = 0$  otherwise.

- The **modified regression algorithm** tries to update one column/row of  $\Theta$  at a time.
- Let  $\Theta^{(t)}$  be the current estimate, and  $\mathbf{W}^{(t)} = [\Theta^{(t)}]^{-1}$ . Will show how to get  $\Theta^{(t+1)}$ .
- W.L.O.G., will show how to update the last column/row of  $\Theta$ .



# Modified Regression Algorithm

- The algorithm relies on the identity

$$\begin{pmatrix} \mathbf{W}_{11}^{(t)} & \mathbf{w}_{12} \\ \mathbf{w}_{12}' & w_{22} \end{pmatrix} \begin{pmatrix} \Theta_{11}^{(t)} & \boldsymbol{\theta}_{12} \\ \boldsymbol{\theta}_{12}' & \theta_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0}' & 1 \end{pmatrix}.$$

- Will try to update  $\boldsymbol{\theta}_{12}$  and  $\theta_{22}$  such that the last row/column of the gradient condition is fulfilled.
- The vector  $\mathbf{w}_{12}$  is linked to  $\boldsymbol{\theta}_{12}$  and  $\theta_{22}$  through

$$\mathbf{W}_{11}^{(t)} \boldsymbol{\theta}_{12} + \theta_{22} \mathbf{w}_{12} = \mathbf{0}.$$

- $\boldsymbol{\theta}_{12}$  has to satisfy the edge constraint.
- Let  $\boldsymbol{\beta} := -\boldsymbol{\theta}_{12}/\theta_{22}$ , then  $\mathbf{w}_{12} = \mathbf{W}_{11}^{(t)} \boldsymbol{\beta}$ , and

$$\mathbf{W}_{11}^{(t)} \boldsymbol{\beta} - \mathbf{s}_{12} - \boldsymbol{\gamma}_{12} = \mathbf{0}. \tag{1}$$

- Note that  $\boldsymbol{\beta}$  has the same zero coordinates as  $\boldsymbol{\theta}_{12}$ .
- The last diagonal element  $w_{22}$  has to equal to  $s_{22}$ , and

$$\mathbf{w}_{12}' \boldsymbol{\theta}_{12} + w_{22} \theta_{22} = 1.$$

- More details in class demonstration.

# Modified Regression Algorithm

---

**Algorithm 17.1** *A Modified Regression Algorithm for Estimation of an Undirected Gaussian Graphical Model with Known Structure.*

---

1. Initialize  $\mathbf{W} = \mathbf{S}$ .
  2. Repeat for  $j = 1, 2, \dots, p, 1, \dots$  until convergence:
    - (a) Partition the matrix  $\mathbf{W}$  into part 1: all but the  $j$ th row and column, and part 2: the  $j$ th row and column.
    - (b) Solve  $\mathbf{W}_{11}^* \beta^* - s_{12}^* = 0$  for the unconstrained edge parameters  $\beta^*$ , using the reduced system of equations as in (17.19). Obtain  $\hat{\beta}$  by padding  $\hat{\beta}^*$  with zeros in the appropriate positions.
    - (c) Update  $w_{12} = \mathbf{W}_{11} \hat{\beta}$
  3. In the final cycle (for each  $j$ ) solve for  $\hat{\theta}_{12} = -\hat{\beta} \cdot \hat{\theta}_{22}$ , with  $1/\hat{\theta}_{22} = s_{22} - w_{12}^T \hat{\beta}$ .
-

# Graphical Lasso

- To estimate the graph structure, consider maximizing the penalized log-likelihood

$$\log \det(\Theta) - \text{trace}(\mathbf{S}\Theta) - \lambda \|\Theta\|_1,$$

where  $\|\Theta\|_1$  denotes the sum of the absolute values of the elements of  $\Theta$ .

- The gradient condition becomes

$$\Theta^{-1} - \mathbf{S} - \lambda \cdot \text{Sign}(\Theta) = \mathbf{0},$$

- Here  $\text{Sign}(x)$  is defined as the sub-differential of  $|x|$ , which is the sign when  $x \neq 0$ , and can take any value in  $[-1, 1]$  when  $x = 0$ .
- Similar to the Modified Regression Algorithm, update one column/row of  $\Theta$  at a time.
- The gradient condition (1) becomes

$$\mathbf{W}_{11}^{(t)} \boldsymbol{\beta} - \mathbf{s}_{12} - \lambda \cdot \text{Sign}(\boldsymbol{\beta}) = \mathbf{0}.$$

- Can adapt Lasso to solve this optimization problem, using the pathwise coordinate descent method. More details in the class.

## Example

- Read

<https://people.math.aau.dk/~sorenh/misc/2012-Oslo-GMwR/GMwR-notes.pdf>

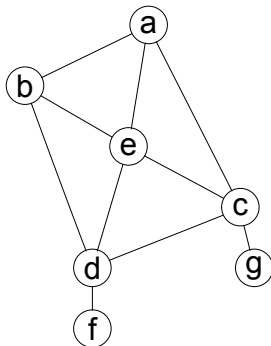
- Run `setRepositories()`, and make sure that **BioC software** is added.
- Install the following packages using the command

```
install.packages("gRbase", dependencies=TRUE)
install.packages("gRain", dependencies=TRUE)
install.packages("gRim", dependencies=TRUE)
install.packages("Rgraphviz", dependencies=TRUE)
```

## Example: Generate and plot an undirected graph

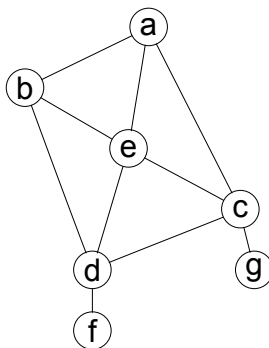
```
library(gRbase)
library(Rgraphviz)

g1 <- ug(~a:b:e + a:c:e + b:d:e + c:d:e + c:g + d:f)
as(g1, "matrix")
## a b e c d g f
## a 0 1 1 1 0 0 0
## b 1 0 1 0 1 0 0
## e 1 1 0 1 1 0 0
## c 1 0 1 0 1 1 0
## d 0 1 1 1 1 0 0 1
## g 0 0 0 0 1 0 0 0
## f 0 0 0 0 0 1 0 0
plot(g1)
```



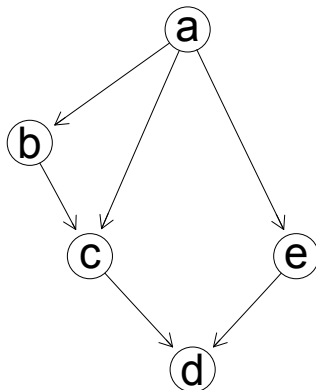
## Example: Using adjacency matrix

```
m <- matrix(c(0,1,1,0,1,1,0,0,1,1,1,0,0,1,1,0,1,1,1,1,1,0), nrow = 5)
rownames(m) <- colnames(m) <- c("a", "b", "c", "d", "e")
m
g1=as(m, "graphNEL")
g1
## A graphNEL graph with undirected edges
## Number of Nodes = 11
## Number of Edges = 8
plot(g1)
```



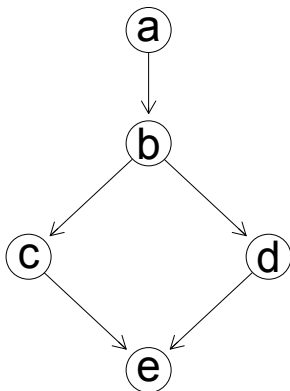
## Example: Generate and plot a directed graph

```
dag0 <- dag(~a, ~b * a, ~c * a * b, ~d * c * e, ~e * a)
dag0 <- dag(~a + b:a + c:a:b + d:c:e + e:a)
dag0 <- dag("a", c("b", "a"), c("c", "a", "b"), c("d", "c", "e"), c("e", "a"))
dag0
## A graphNEL graph with directed edges
## Number of Nodes = 5
## Number of Edges = 6
plot(dag0)
```



## Example: Using directed adjacency matrix

```
m <- matrix(c(0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,1,1,0), nrow = 5)
rownames(m) <- colnames(m) <- letters[1:5]
m
## 1 indicates an arrow from row to column
##   a b c d e
## a 0 1 0 0 0
## b 0 0 1 1 0
## c 0 0 0 0 1
## d 0 0 0 0 1
## e 0 0 0 0 0
dag1 <- as(m, "graphNEL")
plot(dag1)
```





## Example: Protein flow cytometry data

`https://web.stanford.edu/~hastie/ElemStatLearn/datasets/sachs.data`

7466 cells, one per row

11 proteins, one per column

Protein names are:

"praf",  
"pmek",  
"plcg",  
"PIP2",  
"PIP3",  
"p44/42",  
"pakts473",  
"PKA",  
"PKC",  
"P38",  
"pjnk"

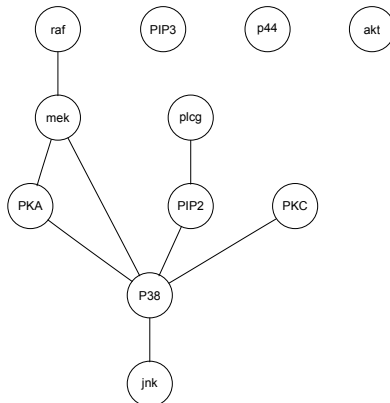
Sachs, K., and Perez, O., and Pe'er, D., and Lauffenburger, D. and Nolan, G.  
Causal protein-signaling networks derived from multiparameter single-cell data  
Science, 308(5721), 523--529, 2005.

## Example: Protein flow cytometry data

```
pfc=read.table("sachs.txt",header=F)
colnames(pfc)=c(
  "raf",
  "mek",
  "plcg",
  "PIP2",
  "PIP3",
  "p44",
  "akt",
  "PKA",
  "PKC",
  "P38",
  "jnk")
dim(pfc)
S=cov(pfc)
dim(S)
S=S/1000 ## rescale the sample covariance matrix
```

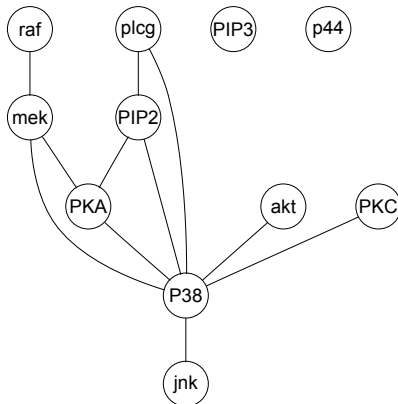
## Example: $\lambda = 36$

```
fit1=glasso(S,rho=36)
Theta=fit1$wi
colnames(Theta)=c("raf","mek","plcg","PIP2","PIP3","p44","akt","PKA","PKC","P38","jnk")
rownames(Theta)=c("raf","mek","plcg","PIP2","PIP3","p44","akt","PKA","PKC","P38","jnk")
Adj=Theta!=0
Adj=Adj*1
diag(Adj)=0
g1=as(Adj, "graphNEL")
plot(g1)
```



## Example: $\lambda = 27$

```
fit2=glasso(S,rho=27)
Theta=fit2$wi
colnames(Theta)=c("raf","mek","plcg","PIP2","PIP3","p44","akt","PKA","PKC","P38","jnk")
rownames(Theta)=c("raf","mek","plcg","PIP2","PIP3","p44","akt","PKA","PKC","P38","jnk")
Adj=Theta!=0
Adj=Adj*1
diag(Adj)=0
g2=as(Adj, "graphNEL")
plot(g2)
```



## Example: $\lambda = 7$

```
fit3=glasso(S,rho=7)
Theta=fit3$wi
colnames(Theta)=c("raf","mek","plcg","PIP2","PIP3","p44","akt","PKA","PKC","P38","jnk")
rownames(Theta)=c("raf","mek","plcg","PIP2","PIP3","p44","akt","PKA","PKC","P38","jnk")
Adj=Theta!=0
Adj=Adj*1
diag(Adj)=0
g3=as(Adj, "graphNEL")
plot(g3)
```

