

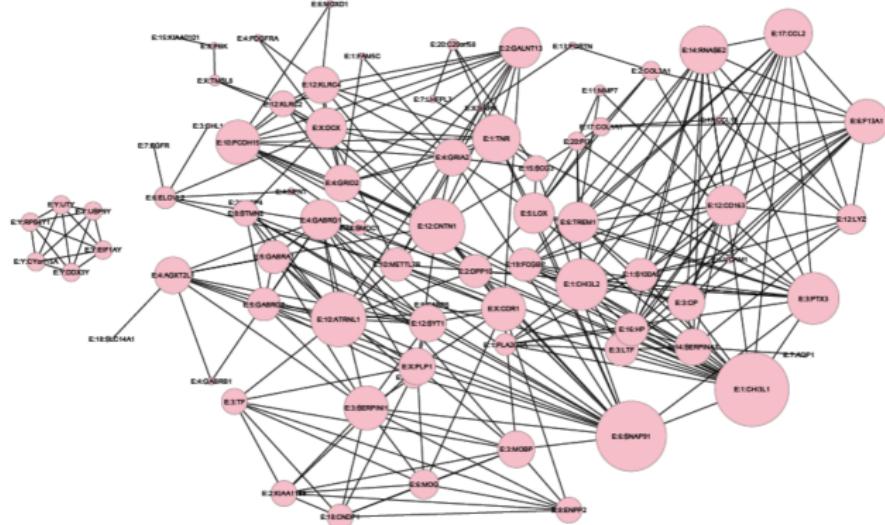
# Unsupervised Learning: Graphical Models

# Networks



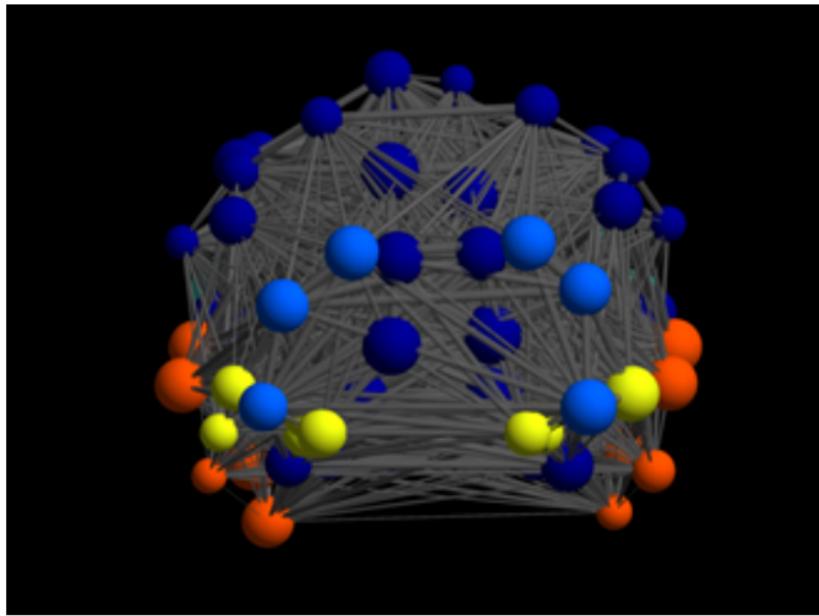
Depicts relationships (edges) between features (nodes).

## Networks



## Genomics: Relationships between genes.

# Networks



Neuroimaging: Relationships between brain regions.

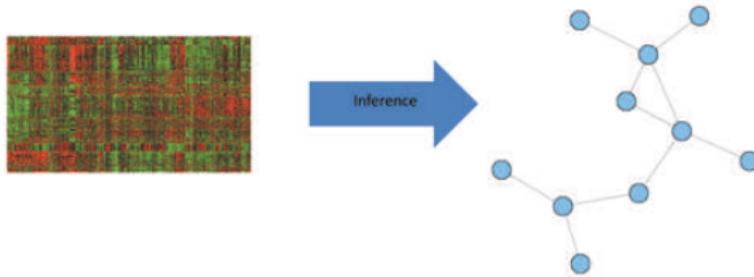
# Networks in Unsupervised Learning

## ① Network Data.

- ▶ Examples: Social networks, twitter, citations, surveillance, web links, etc.

## ② Our focus: Learn network from data (structural network learning).

- ▶ Data matrix:  $X_{n \times p}$ .
- ▶ Features form  $p$  nodes.
- ▶ Goal: Learn edges between nodes  $\equiv$  learn the relationships between features.



# Network Data

- Network data have arisen as one of the most common forms of information collection.
- Networks consist of two components:
  - ① nodes or vertices corresponding to basic units of a system;
  - ② edges representing connections between nodes.
- Examples:
  - ① Nodes might be humans in social networks; molecules, genes, or neurons in biology networks, or web pages in information networks.
  - ② Edges could be friendships, alliances, URLs, or citations.
- A network can be represented by an adjacency matrix.

## Stochastic Block Models

- There are different statistical models for network data, and the stochastic block model (SBM) is very popular.
- In a SBM, the nodes are partitioned into  $K < n$  disjoint groups, or *communities*, according to some latent random mechanism.
- A stochastic block model with  $n$  nodes and  $K$  communities is parameterized by a pair of matrices  $(\Theta, B)$ , where  $\Theta \in \mathbb{M}_{n,K}$  is the membership matrix (each row has exactly one 1 and  $(K - 1)$  0's) and  $B \in \mathbb{R}^{K \times K}$  is a symmetric connectivity matrix with edge probabilities.
- The adjacency matrix  $A = (a_{ij})_{1 \leq i,j \leq n}$  is generated as

$$a_{ij} = \begin{cases} \text{independent Bernoulli } (B_{g_i g_j}), & \text{if } i \leq j \\ a_{ji}, & \text{if } i > j \end{cases}$$

where  $g_i$  ( $1 \leq g_i \leq K$ ) is the community label.

- Variants of block models: degree-corrected SBM, dynamic SBM, etc.

# Community Detection

- The basic goal of community detection is to partition the vertices of a graph into clusters that are more densely connected.
- In SBMs, community detection aims to recover the membership matrix  $\Theta$  up to column permutations.
- Assuming  $K$  is known, different procedures are proposed to solve the problem, including likelihood methods, spectral clustering, modularity maximization, etc.

# Community Detection and Stochastic Block Models

Example of one spectral clustering algorithm in SBM (Lei and Rinaldo, 2015)

---

**Algorithm 1:** Spectral clustering with approximate  $k$ -means

---

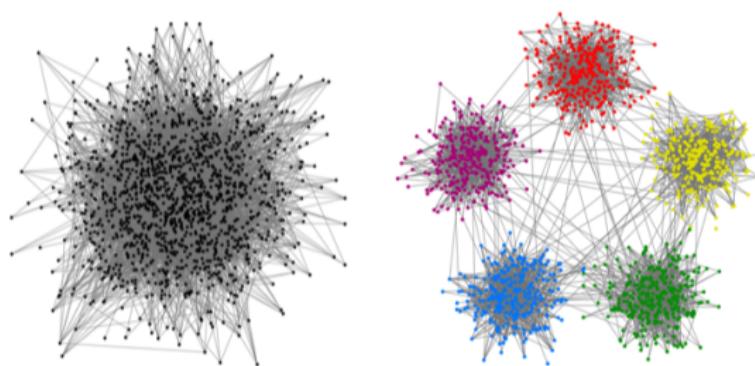
**Input** : Adjacency matrix  $A$ ; number of communities  $K$ ;  
approximation parameter  $\varepsilon$ .

**Output:** Membership matrix  $\widehat{\Theta} \in \mathbb{M}_{n,K}$ .

1. Calculate  $\widehat{U} \in \mathbb{R}^{n \times K}$  consisting of the leading  $k$  eigenvectors  
(ordered in absolute eigenvalue) of  $A$ .
  2. Let  $(\widehat{\Theta}, \widehat{X})$  be an  $(1 + \varepsilon)$  -approximate solution to the  $k$  -means  
problem with  $K$  clusters and input matrix  $\widehat{U}$ .
  3. Output  $\widehat{\Theta}$ .
-

# Community Detection and Stochastic Block Models

- The following two graphs are the same graph re-organized and drawn from the SBM model with 1000 vertices, 5 balanced communities, within-cluster probability of  $1/50$  and across-cluster probability of  $1/1000$ .
- The goal of community detection in this case is to obtain the right graph (with the true communities) from the left graph (scrambled) up to some level of accuracy.



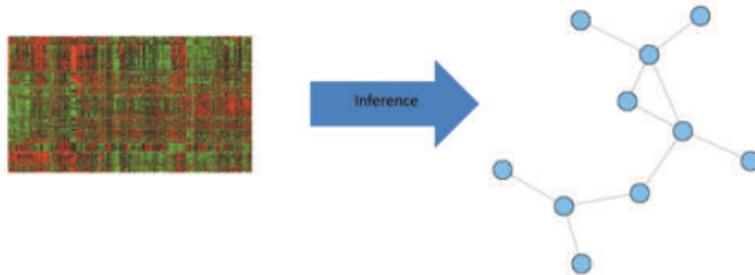
# Networks in Unsupervised Learning

## ① Network Data.

- ▶ Examples: Social networks, twitter, citations, surveillance, web links, etc.

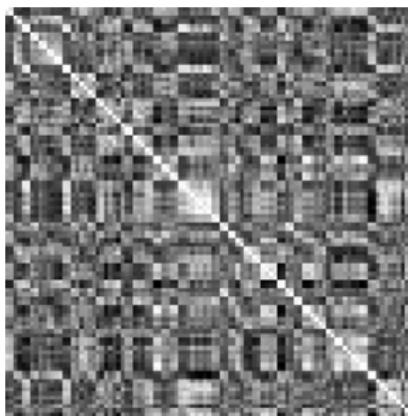
## ② Our focus: Learn network from data (structural network learning).

- ▶ Data matrix:  $X_{n \times p}$ .
- ▶ Features form  $p$  nodes.
- ▶ Goal: Learn edges between nodes  $\equiv$  learn the relationships between features.



# Correlation Networks (Association Networks)

- Simplest (and most-widely used!) method for estimating networks; assumes that edges correspond to large correlation magnitudes.
- Let  $r(i, j)$  be correlation between  $X_i$  and  $X_j$ ; we claim an **edge between  $i$  and  $j$**  if  $|r(i, j)| > \tau$ .
  - ▶  $\tau$ : a user-specified threshold (**tuning parameter**).



Correlation matrix.



Thresholded correlation matrix.

# Limitations of Correlation Networks

- ① The estimation is highly dependent on the choice of  $\tau$ .
- ② Correlation captures linear associations, but many real-world relationships are nonlinear.
- ③ Large correlations can occur due to confounding.

# Limitations of Correlation Networks

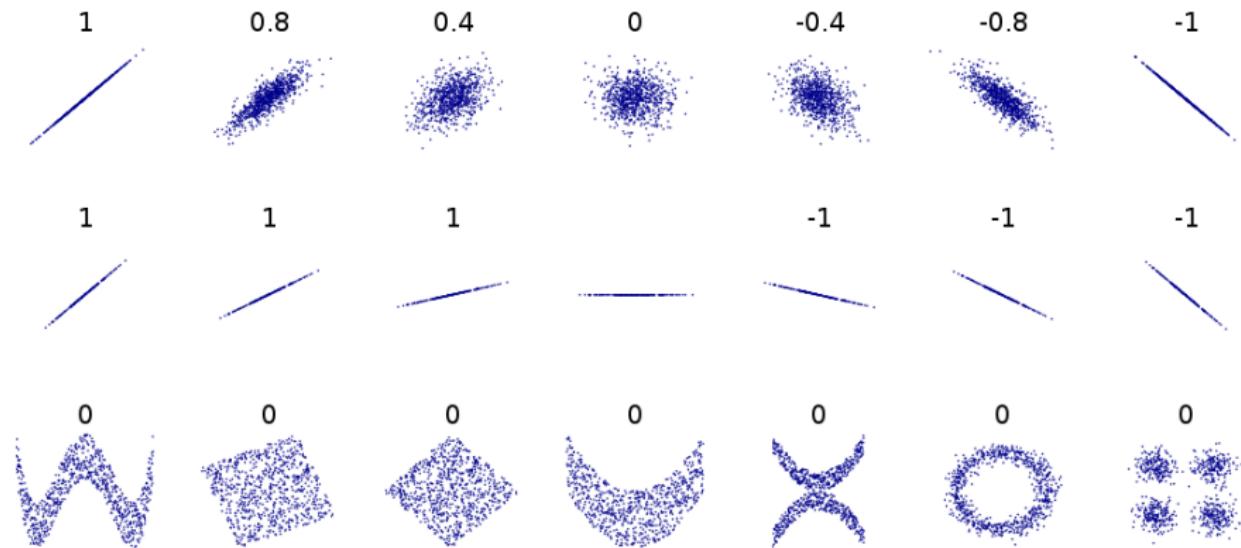
The estimation is highly dependent on the choice of  $\tau$ .

- We can instead test  $H_0 : r_{xy} = 0$
- A commonly used test is given by the Fisher transformation

$$Z = \frac{1}{2} \ln \left( \frac{1+r}{1-r} \right) = \text{artanh}(r) \sim_{H_0} N \left( 0, \frac{1}{\sqrt{n-3}} \right)$$

# Limitations of Correlation Networks

Correlation captures **linear associations**, but many real-world relationships are **nonlinear**.



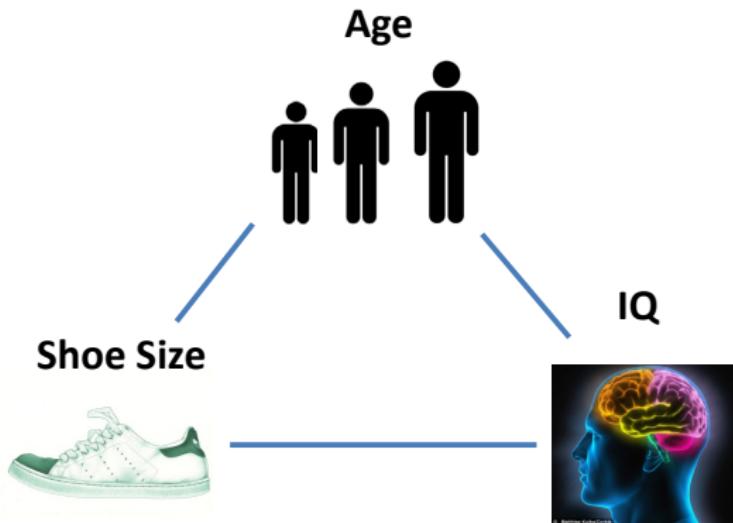
# Limitations of Correlation Networks

Correlation captures **linear associations**, but **many real-world relationships are nonlinear**.

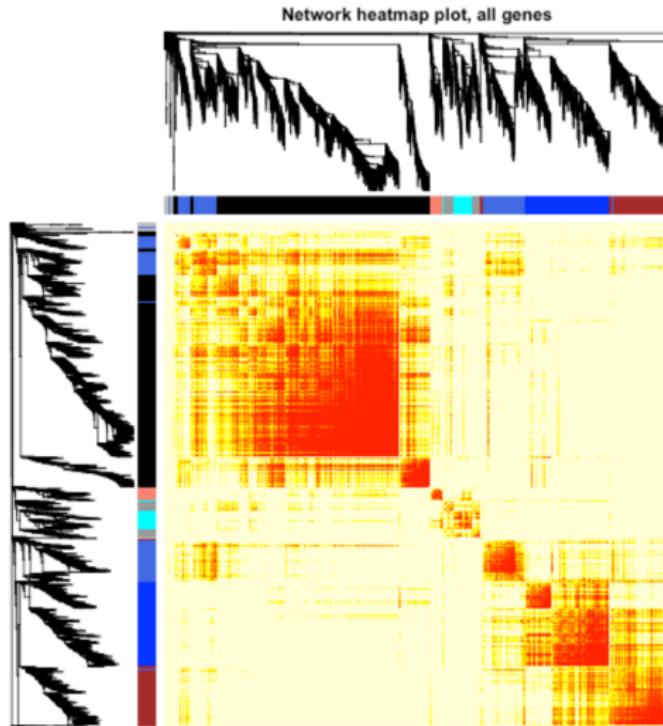
- We can use other measures of association, for instance, **Spearman correlation** or **Kendal's  $\tau$** .
  - ▶ These methods define correlation between two variables, based on the **ranking** of observations, and not their exact values.
  - ▶ They can better capture non-linear associations.
- We can instead use **mutual information**; this has been used in many algorithm, e.g. ARACNE.

# Limitations of Correlation Networks

Large correlations can occur due to **confounding**.



# Correlation Network Example: WGCNA



**Weighted Gene Co-Expression Network Analysis**  
Popular R Package

# Correlation Network Example: WGCNA

WGCNA Approach:

- ① Thresholds the correlation matrix.
- ② Raises this to an even-integer power (usually 6).
- ③ Measures modularity via a topological approach.
- ④ Finds modules (clusters) using hierarchical clustering of co-modularity.
- ⑤ Analyzes modules.

Strengths?

Weaknesses?

# Graphical Models

## Probabilistic Graphical Models

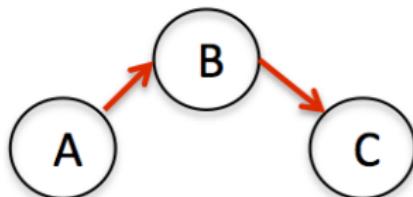
Joint multivariate probability distribution where dependencies can be represented as a network.

Advantages:

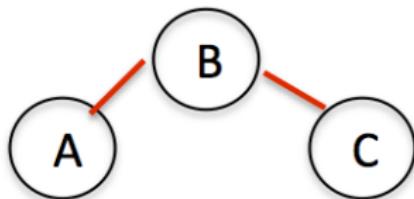
- Graphical models offer efficient factorized forms for joint distributions with easily interpretable dependencies.
  - ▶ **Conditional dependencies** denoted via an edge in network.
- Convenient visual representation.

# Two Major Types of Graphical Models

- ① Directed - Bayesian Networks.
  - ▶ Directed Acyclic Graphs.



- ② Undirected - Markov Networks.
  - ▶ **Our Focus!**

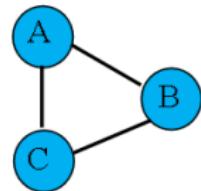


# Markov Networks

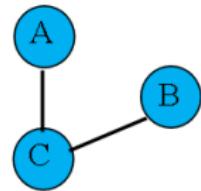
## Markov Network

An *undirected graphical model* that characterizes **conditional dependence** ( $\equiv$  direct relationships).

- *Edge*: Two nodes are **conditionally dependent**.
- *No edge*: Two nodes are **conditionally independent**.
- Conditions on all other nodes.



$$A \perp B \mid C$$

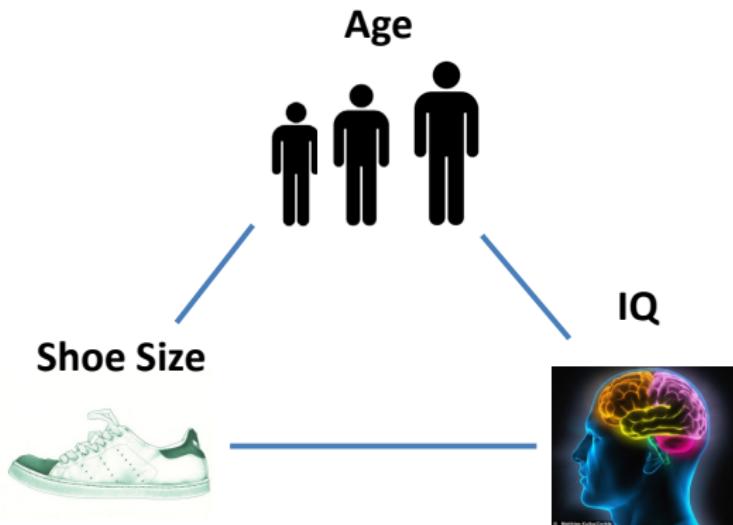


# Markov Networks — Conditional Dependence

Regression Interpretation:

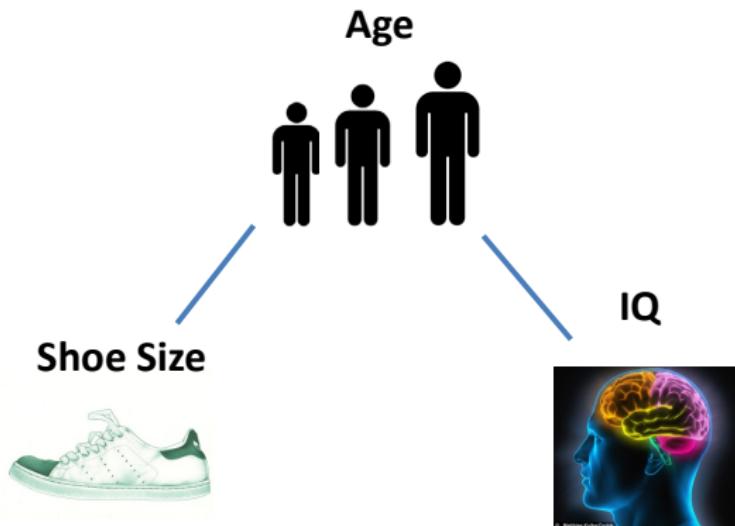
- Imagine trying to predict the observations in **Node A** (response) by the observations of all other nodes (predictors).
- **Node B** predictive of **Node A** (with all other nodes in model).
  - ▶ **A** is conditionally dependent on **B**.
  - ▶ Edge.
- Because of other nodes in model, **Node B** does not add any predictive value for **Node A**.
  - ▶ **A** is conditionally independent of **B**.
  - ▶ No Edge.

# Markov Networks — Conditional Dependence



Correlation.

# Markov Networks — Conditional Dependence



Conditional Dependence.

# Markov Networks — Conditional Dependence

How can we learn conditional dependencies?

- $A$  and  $B$  are conditionally independent given  $C$  if

$$P(A, B \mid C) = P(A \mid C)P(B \mid C)$$

- ▶ Generally difficult (need to estimate multivariate densities).
- Alternatively, can use nonparametric approaches, e.g. **conditional mutual information**, but not easy in high dimensions.
- Often resort to models, or simple measures, such as **partial correlations**...

# Partial Correlation

- Partial correlation measures the correlation between  $A$  and  $B$  after the effect of the other variables are removed.
  - ▶ In our example, this means correlation between shoe size and IQ, after adjusting for age.
- The partial correlation between  $A$  and  $B$  given  $C$  is given by:

$$\rho_{AB \cdot C} \equiv \rho(A, B|C) = \frac{\rho_{AB} - \rho_{AC}\rho_{BC}}{\sqrt{1 - \rho_{AC}^2}\sqrt{1 - \rho_{BC}^2}}.$$

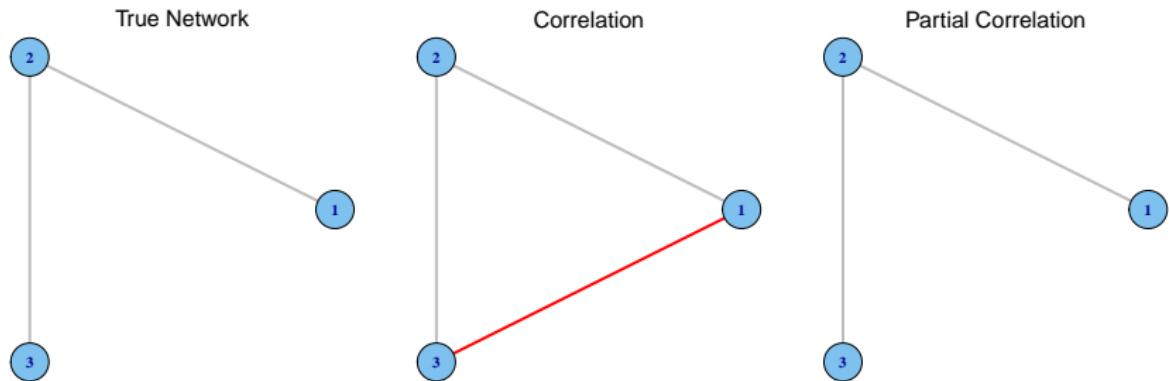
- Alternatively, regress  $A$  on  $C$  and get the residual,  $r_A$ ; do the same for  $B$  to get  $r_B$ . The partial correlation between  $A$  and  $B$  given  $C$  is  $\text{Cor}(r_A, r_B)$ .

# Partial Correlation

- Partial correlation is **symmetric**  $\Rightarrow$  **undirected network**
- Partial correlation is a number **between -1 and 1**
- In partial correlation networks, we **draw an edge** between  $A$  and  $B$ , **if the partial correlation between them is large**
- Calculation of partial correlation is more difficult

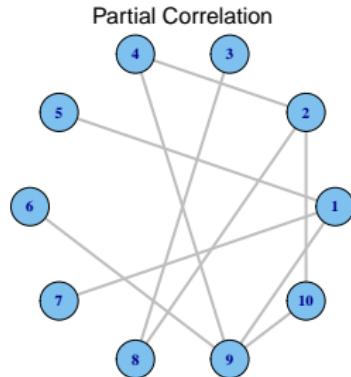
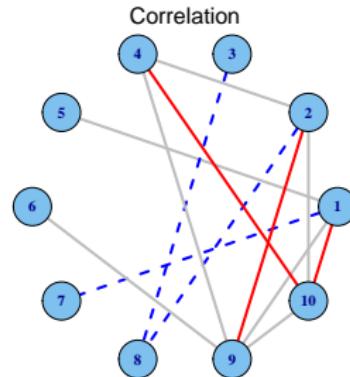
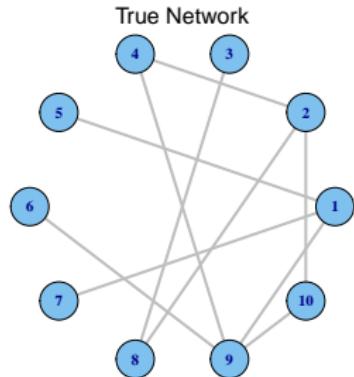
# A Simple Example

$$\text{Correlation} = \begin{bmatrix} 1 & -.8 & .7 \\ -.8 & 1 & -.8 \\ .7 & -.8 & 1 \end{bmatrix} \quad \text{PartialCorr} = \begin{bmatrix} 1 & .6 & 0 \\ .6 & 1 & .6 \\ 0 & .6 & 1 \end{bmatrix}$$



# A Larger Example

- A network with 10 nodes and 20 edges
- $n = 100$  observations
- Estimation using correlation & partial correlation (20 edges)

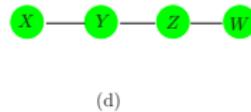
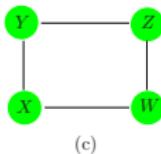
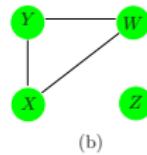
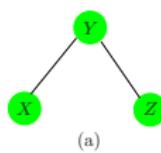


# Gaussian Graphical Models (GGMs)

# Partial Correlation for Gaussian Random Variables

- For Gaussian (multivariate normal) random variables, partial correlation between  $X_i$  and  $X_j$  given all other variables is given by the inverse of the (standardize) covariance matrix  $\Sigma$ .
  - ▶ The  $(i, j)$  entry in  $\Sigma^{-1}$  gives the partial correlation between  $X_i$  and  $X_j$  given all other variables  $X_{\setminus i, j}$ .
- Gaussian Graphical Model (GGM):
  - ▶ Multivariate normal:  $\mathbf{X} \sim N(\mathbf{0}, \Sigma)$
  - ▶  $\Theta = \Sigma^{-1}$  = inverse covariance/precision/concentration matrix.
  - ▶ Zeros in  $\Theta \implies$  conditional independence!
  - ▶ Edges correspond to non-zeros in  $\Theta$ .

# Partial Correlation for Gaussian Random Variables



$$\begin{pmatrix} - & \times & 0 \\ \times & - & \times \\ 0 & \times & - \end{pmatrix}$$

$$\begin{pmatrix} - & \times & \times & 0 \\ \times & - & \times & 0 \\ \times & \times & - & 0 \\ 0 & 0 & 0 & - \end{pmatrix}$$

$$\begin{pmatrix} - & \times & 0 & \times \\ \times & - & \times & 0 \\ 0 & \times & - & \times \\ \times & 0 & \times & - \end{pmatrix}$$

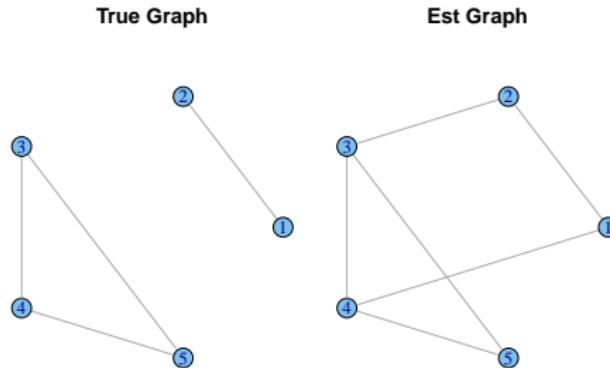
$$\begin{pmatrix} - & 0 & 0 & \times \\ 0 & - & \times & 0 \\ 0 & \times & - & \times \\ \times & 0 & \times & - \end{pmatrix}$$

# Estimating GGMs

From our discussions so far, to estimate the network, we can

- ① Calculate the **empirical covariance matrix**: for (centered)  $n \times p$  data matrix  $X$ ,  $\mathbf{S} = (n - 1)^{-1} X^T X$ .
- ② **Find the inverse of  $\mathbf{S}$** . Non-zero values of  $\mathbf{S}^{-1}$  give the edges.

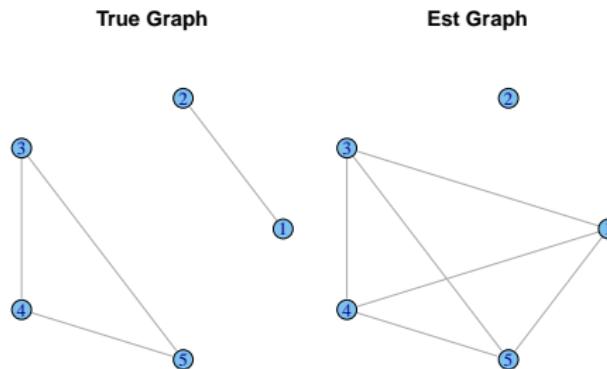
While simple, this may not work well in practice, even with large samples!



# Estimating GGMs in High Dimensions

Many problems arise in high-dimensional settings, when  $p \gg n$ .

- First,  $S$  is not invertible if  $p > n$ !
- Even if  $p < n$ , but  $n$  is not very large, we may still get poor estimates, and many false positives/negatives.



# Estimating GGMs in High Dimensions

- A number of methods have been recently proposed for estimating GGMs in high dimensions.
- The main idea in most of these methods is to **use a regularization penalty**, like the **lasso**.
- We discuss two approaches:
  - ▶ neighborhood selection
  - ▶ graphical lasso

# Estimating GGMs in High Dimensions – Method 1

The idea behind **neighborhood selection**, is to estimate the graph by fitting a **penalized regression of each variable on all other variables**.

- Find **neighbors** of each node  $X_j$  by  $l_1$ -penalized regression or lasso:

$$\underset{\beta^j}{\text{minimize}} \quad \|X_j - \mathbf{X}_{\neq j} \beta^j\|_2^2 + \lambda \sum_{k \neq j} |\beta_k^j|$$

- The final estimate is found by combining all of the edges from these individual regression problems.
  - ▶ Symmetry —  $\beta_k^j$  not always same as  $\beta_j^k$ .
  - ▶ Use min or max rule.

# Estimating GGMs in High Dimensions – Method 2

Estimate a sparse  $\Theta$  via penalized maximum likelihood estimation (MLE).

## Graphical Lasso (glasso)

$$\underset{\Theta}{\text{maximize}} \quad \text{logdet}(\Theta) - \text{tr}(S\Theta) - \lambda \|\Theta\|_1$$

- Blue: Log-likelihood; logdet is log-determinant and tr is matrix trace.
- Red: Penalty that encourages zeros in off-diagonal elements of  $\Theta$ .

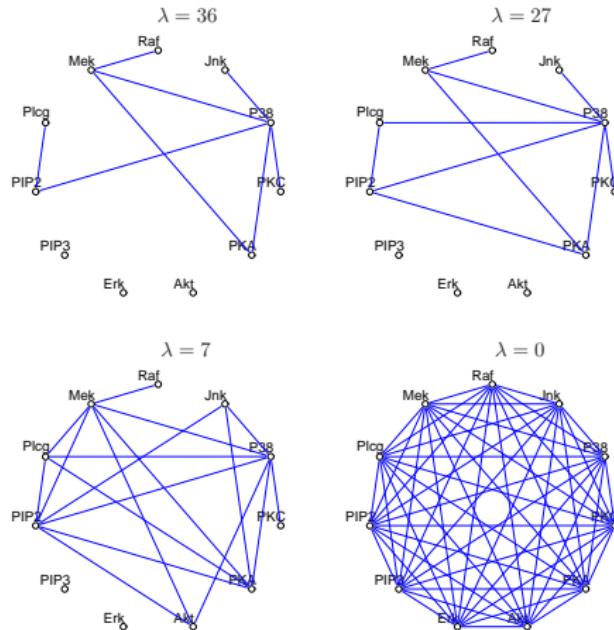
# Comparing the Two Approaches

- Neighborhood selection is an **approximation to the graphical lasso**:
  - ▶ Consider regression of  $X_j$  on  $X_k, j \neq k$
  - ▶ Then the regression coefficient for neighborhood selection is related to the  $j, k$  element of  $\Theta$ :
- Neighborhood selection is computationally more efficient, and may give better graph estimates, but does not give an estimate of  $\Theta$ !

$$\beta_k^j = -\frac{\Theta_{jk}}{\Theta_{jj}}$$

# A Real Example

- Flow cytometry proteomics in single cells (Sachs et al, 2003).
- $p = 11$  proteins measured in  $n = 7466$  cells



# How to Choose $\lambda$ ?

- $\lambda$  modulates trade-off between **model fit** and **network sparsity**:
  - ▶  $\lambda = 0$  gives a dense network (no sparsity).
  - ▶ As  $\lambda$  increases, network becomes more sparse.
- A number of approaches available

- ① **Cross-Validation** — tends to yield overly dense networks.
- ② **Extended BIC** — adjusted BIC for high dimensions.
- ③ **Controlling the probability of falsely connecting disconnected components** at level  $\alpha$  (Banerjee et al, 2008):

$$\lambda(\alpha) = \frac{t_{n-2}(\alpha/2p^2)}{\sqrt{n-2 + t_{n-2}(\alpha/2p^2)}},$$

( $t_{n-2}(\alpha)$  is the  $(100 - \alpha)\%$  quantile of  $t$ -distribution with  $n - 2$  d.f.)

- ④ **Stability selection** — Choose  $\lambda$  that gives the most **stable network** (R: huge package)

# Other Types of Graphical Models

# Nonparanormal (Gaussian Copula) Models

- Suppose  $X \sim N(0, \Sigma)$ , but there exists monotone functions  $f_j, j = 1, \dots, p$  such that  $[f_1(X_1), \dots, f_p(X_p)] \sim N(0, \Sigma)$ 
  - ▶  $X$  has a nonparanormal distribution  $X \sim NPN_p(f, \Sigma)$ .
  - ▶  $f$  and  $\Sigma$  are parameters of the distribution, and estimated from data.
  - ▶ For continuous distributions, the nonparanormal family is the same as the Gaussian copula family
- To estimate the nonparanormal network:
  - i) transform the data:  $[f_1(X_1), \dots, f_p(X_p)]$
  - ii) estimate the network of the transformed data (e.g. calculate the empirical covariance matrix of the transformed data, and apply glasso or neighborhood selection)

## A Related Procedure

- Liu et al (2012) and Xue & Zou (2012) proposed a closely related idea using **rank-based correlation**
  - ▶ Let  $r_j^i$  be the **rank of  $x_j^i$**  among  $x_j^1, \dots, x_j^n$  and  $\bar{r}_j = (n+1)/2$  be the average rank
  - ▶ Calculate **Spearman's  $\rho$**  or **Kendall's  $\tau$**

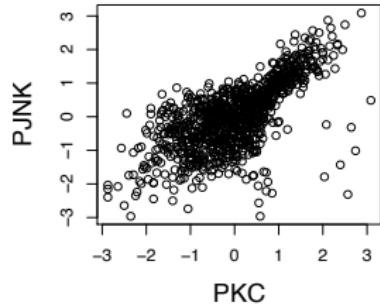
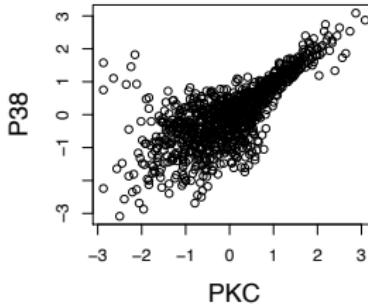
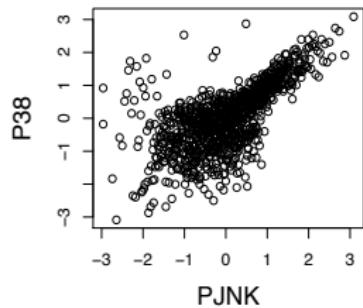
$$\hat{\rho}_{jk} = \frac{\sum_{i=1}^n (r_j^i - \bar{r}_j)(r_k^i - \bar{r}_k)}{\sqrt{\sum_{i=1}^n (r_j^i - \bar{r}_j)^2 \sum_{i=1}^n (r_k^i - \bar{r}_k)^2}}$$

$$\hat{\tau}_{jk} = \frac{2}{n(n-1)} \sum_{1 \leq i < i' \leq n} \text{sign} \left( (x_j^i - x_j^{i'})(x_k^i - x_k^{i'}) \right)$$

- If  $X \sim NPN_p(f, \Sigma)$ , then  $\Sigma_{jk} = 2 \sin(\rho_{jk}\pi/6) = \sin(\tau_{jk}\pi/2)$
- Therefore, we can estimate  $\Sigma^{-1}$  by **plugging in rank-based correlations into graphical lasso** (R-package `huge`)

# A Real Data Example

- Protein cytometry data for cell signaling (Sachs et al, 2005)
- Transform the data using **Gaussian copula** (Liu et al, 2009), giving marginal normality
- Pairwise relationships still seem **non-linear**



- Shapiro-Wilk test rejects multivariate normality:  $p < 2 \times 10^{-16}$

# Graphical Models for Discrete Random Variables

- In many cases, biological data are not Gaussian: SNPs, RNAseq, etc
- Need to estimate CIG for other distributions: **binomial**, **poisson**, etc
- In this case, the estimators do not have a closed-form!
- A special case, which is computationally more tractable, is the class of **pairwise MRFs**

# Pairwise Markov Random Fields

- The idea of pairwise MRFs is to “assume” that **only two-way interactions among variables** exist
  - ▶ The pairwise MRF associated with the graph  $G$  over the random vector  $X$  is the family of probability distributions  $P(X)$  that can be written as

$$P(X) \propto \exp \sum_{(j,k) \in E} \phi_{jk}(x_j, x_k)$$

- ▶ For each edge  $(j, k) \in E$ ,  $\phi_{jk}$  is called the **edge potential function**
- For discrete random variables, any MRF can be transformed to an MRF with pairwise interactions by introducing additional variables<sup>1</sup>

---

<sup>1</sup>Wainwright & Jordan, 2008

# Graphical Models for Binary Random Variables

- Suppose  $X_1, \dots, X_p$  are binary random variables, e.g. corresponding to SNPs, or DNA methylation
- A special case of discrete graphical models is the **Ising model for binary random variables**

$$P_\theta(x) = \frac{1}{Z(\theta)} \exp \left\{ \sum_{(j,k) \in E} \theta_{jk} x_j x_k \right\}$$

- ▶ A **pairwise MRF** for binary data, with  $\phi_{jk}(x_j, x_k) = \theta_{jk} x_j x_k$
- ▶  $x^i \in \{-1, +1\}^p$
- ▶ The **partition function**  $Z(\theta)$  ensures that distribution sums to 1
- ▶  $(j, k) \in E$  iff  $\theta_{jk} \neq 0$ !

# Graphical Models for Binary Random Variables

- We can consider a **neighborhood selection**<sup>2</sup> approach with an  $\ell_1$  penalty to find the neighborhood of each node  
 $N(j) = \{k \in V : (j, k) \in E\}$
- For  $j = 1, \dots, p$ , need to solve (after some algebra)

$$\min_{\theta} \left\{ n^{-1} \sum_{i=1}^n \left[ f(\theta; x^i) - \sum_{k \neq j} \theta_{jk} x_j^i x_k^i + \lambda \|\theta_{-j}\|_1 \right] \right\}$$

$$\blacktriangleright f(\theta; x) = \log \left\{ \exp \left( \sum_{k \neq j} \theta_{jk} x_k \right) + \exp \left( - \sum_{k \in -j} \theta_{jk} x_k \right) \right\}$$

- This is equivalent to **solving  $p$  penalized logistic regression** problems, which is straightforward (R-package `glmnet`)

---

<sup>2</sup>Ravikumar et al (2010)

# Other Non-Gaussian Distributions

- Assume a pairwise graphical model

$$P(X) \propto \exp \left\{ \sum_{j \in V} \theta_j \phi_j(X_j) + \sum_{(j,k) \in E} \theta_{jk} \phi_{jk}(X_j, X_k) \right\}$$

- Then, similar to the Ising model, graphical models can be learned for other members of the exponential family
  - Poisson graphical models (for e.g. RNAseq), Multinomial graphical models, etc
  - All of these can be learned using a neighborhood selection approach, using the `glmnet` package<sup>3</sup>
  - We can even learn networks with multiple types of nodes (gene expression, SNPs, and CNVs)<sup>4</sup>

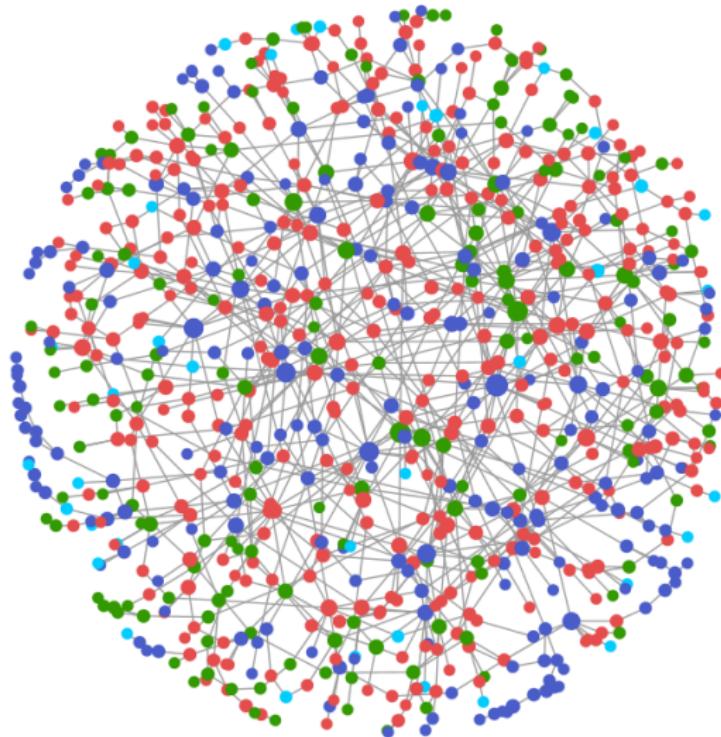
---

<sup>3</sup>Yang et al (2012)

<sup>4</sup>Yang et al (2014), Chen et al (2015)

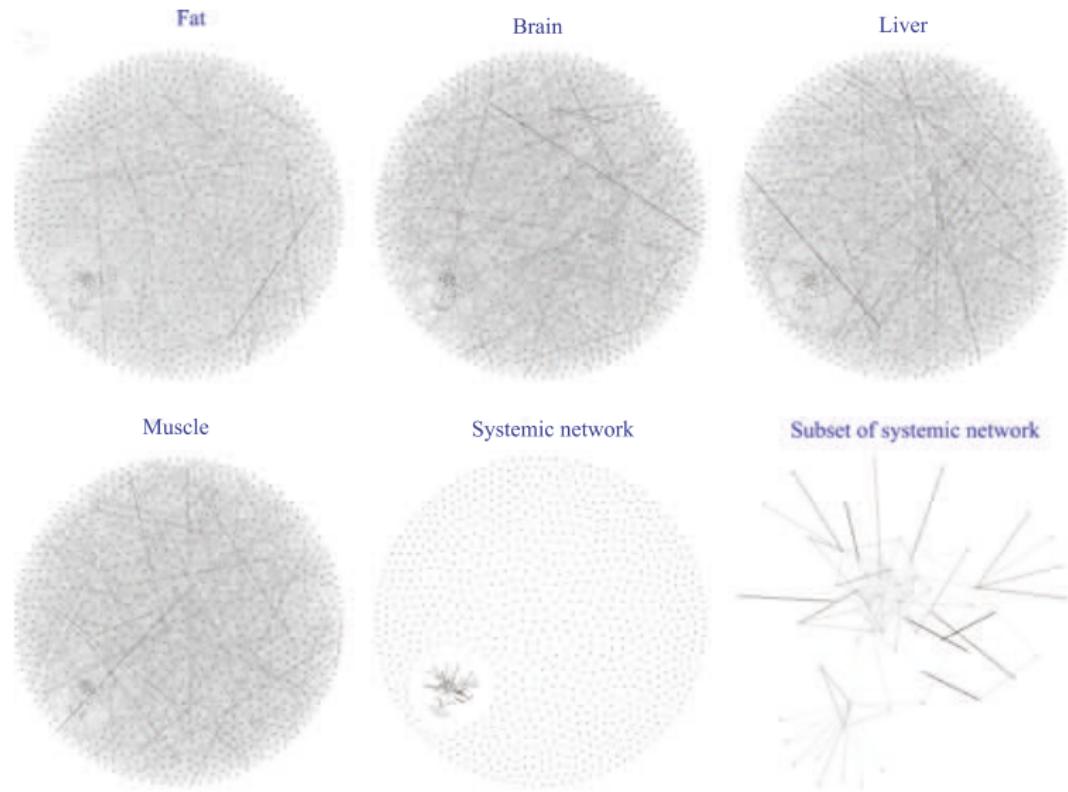
# Research Highlight

## Mixed Graphical Models



# Research Highlight

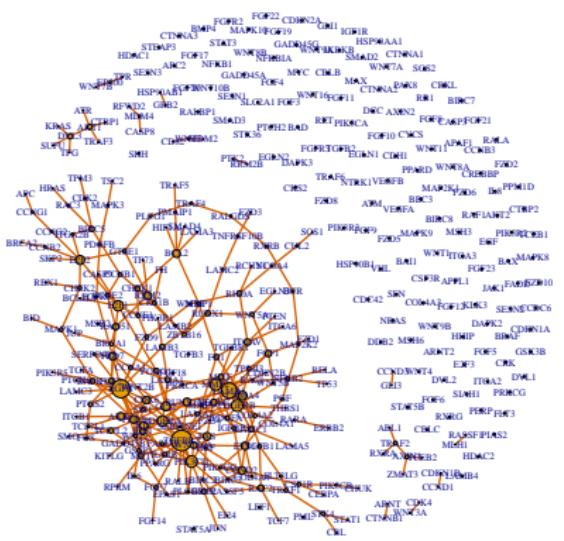
## Multiple Gaussian Graphical Models



# Research Highlight

## Testing Differences in Networks

### Edges in the Positive but not the Negative Group



### Edges in the Negative but not the Positive Group

