STATS 235: Modern Data Analysis
Undirected Graphical Models[1]

Babak Shahbaba

Department of Statistics, UCI

[1] See Hastie el. al. for more details

Graph

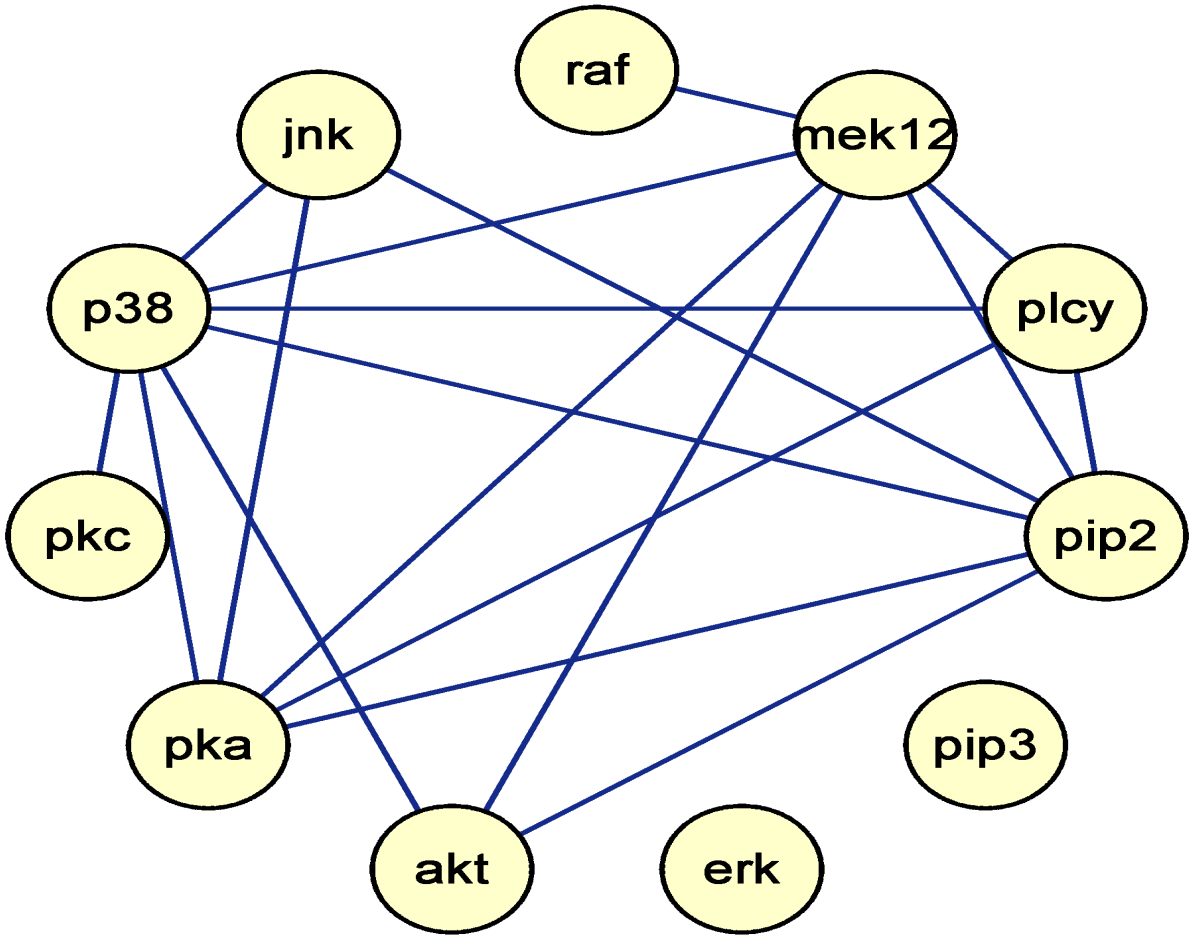


Fig1.11 in Murphy (2012): an undirected graph for 11 proteins.

Graph

- A graph $\mathcal{G}$ consists of a set of vertices V (nodes) and a set of edges E, where each edge connects a pair of nodes.
- Each node represent a variable.
- A graph is a simple representation of the joint distribution of these variables.
- Here, we focus on *undirected* graphs, where edges have no direction.
- Such graphs are also knows as *Markov random fields* or *Markov networks*
- In these graphs, the absence of an edge between two nodes (variables) means *conditional independence* given all other variables.
- In the protein network shown above, Here, jnk and raf and conditionally independent given all other proteins:

$$\text{jnk} \perp \text{raf} \mid \text{rest}$$

Basic properties

- **Adjunct:** Two vertices X and Y are adjunct, $X \sim Y$, if there is an edge joining them:
  $\texttt{p38} \sim \texttt{plcy}$.

- **Path:** A set of vertices $X_1, X_2, \ldots, X_p$ form a path if $X_{i-1} \sim X_i$ for $i = 2, \ldots, p$

- **Complete graph**: A graph where every pair of vertices are adjunct.

- **Subgraph:** a subset of vertices $U \in V$ with the corresponding edges.

Basic properties

- **Pairwise Markov independencies:** The set of all conditional independence relationships in a graph.
- **Separation:** For three subgraphs A, B, and C, we say C *separates* A and B if any path between them intersects a node in C, e.g., `p38` separates `jnk` and `pkc`. In such cases,

$$A \perp B | C$$

  which are called **the global Markov properties** of the graph.
- For a graph, the pairwise and global Markov properties are equivalent so their corresponding probability distributions are the same.

Basic properties

- **Clique:** A complete subgraph, e.g., {p38, jnk, pka}.
- **maximal subgraph:** A clique that we cannot add any other vertices to it while still keeping it as a clique, e.g., {p38, jnk, pka, pip2}.
- We can decompose a complex graph into its set of maximal cliques, $\mathcal{C}$, and write the joint probability density function as

$$f(x) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \varphi_c(x_c)$$

  where Z is the normalizing constant (a.k.a., *partition function*) and φ_c are called *clique potentials*.
- We typically focus on *Pairwise Markov graphs*, where potential functions with at most second-order interactions, $\varphi(x, y)$, are included.

Continuous variables

Undirected graphical models with continuous variables

- Because of convenience, we assume that the variables have multivariate normal distribution $N(\mu, \Sigma)$.
- Note that this is automatically a pairwise Markov graph since Gaussian distribution represents at most second-order interactions.
- The inverse covariance matrix Σ^{-1} captures all the conditional independencies: if $\Sigma^{-1}_{jk} = 0$, then $X_j \perp X_k | \text{rest}$.
- We can use multiple linear regression models to learn the structure of $\Theta = \Sigma^{-1}$.

Parameter estimation given the network structure

- For complete graphs (fully connected), we can maximize the log-likelihood function (up to a constant),

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

where S is the empirical covariance matrix,

$$S = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^\top$$

- In this case, the MLE of Σ is simply S.

Parameter estimation given the network structure

- When some edges are missing, the corresponding elements in $\Theta = \Sigma^{-1}$ are equal to zero.
- This leads to an equality-constrained convex optimization.
- Using Lagrange multipliers γ, we can write the objective function as follows:

$$\ell_C(\Theta) = \log \det\Theta - \mathrm{trace}(S\Theta) - \sum_{(j,k)\notin E} \gamma_{jk}\theta_{jk}$$

with the gradient equation:

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

- Note that the gradient of $\log \det\Theta$ is Θ^{-1}, and the matrix of Lagrange multipliers Γ has nonzero elements for missing edges.

Parameter estimation given the network structure

- Alternatively, as shown in Hastie et. al. (2009), we can iteratively use regression models to estimate Θ and its inverse, $\Omega = \Theta^{-1}$, one variable at a time.

- Consider the *last* variable (which could be any variable since the order does not matter); then, we can decompose Ω and Θ to submatrices

$$\begin{pmatrix} \Omega_{11} & \omega_{12} \\ \omega_{12}^\top & \omega_{22} \end{pmatrix} \begin{pmatrix} \Theta_{11} & \theta_{12} \\ \theta_{12}^\top & \theta_{22} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0^\top & 1 \end{pmatrix}$$

  which means

$$\begin{aligned} \omega_{12} &= -\Omega_{11}\theta_{12}/\theta_{22} \\ &= \Omega_{11}\beta \end{aligned}$$

  where $\beta = -\theta_{12}/\theta_{22}$

- We decompose S in a similar way to create S_{11}, s_{12}, and s_{22}.

Parameter estimation given the network structure

- From the upper right block of the gradient equation, we have

$$\omega_{12} - s_{12} - \gamma_{12} = 0$$

which can be written in terms of β,

$$\Omega_{11}\beta - s_{12} - \gamma_{12} = 0$$

- If we focus on existing edges only, for which the corresponding elements of γ_{12} are zero, we can have following reduced system of equations

$$\Omega^*_{11}\beta^* - s^*_{12} = 0$$

with the solution

$$\hat{\beta}^* = [\Omega^*_{11}]^{-1} s^*_{12}$$

- Then we can pad it with zeros wherever we had missing edges to obtain $\hat{\beta}$.

Algorithm 17.1 in Hastie et. al. (2009)

Initialize $\Omega = S$

Repeat the following steps until convergence

for j $= 1$ to p **do**

Partition the matrices Ω and S into two parts such that Ω_{11} and S_{11} includes all but the jth row and column.

Solve $\Omega_{11}^{*}\beta^{*} - s_{12}^{*} = 0$ for the unconstrained edge parameters β^{*}.

Obtain $\hat{\beta}$ by padding β^{*} with zeros in the appropriate positions (i.e., where there are missing edges).

Set $\omega_{12} = \Omega_{11}\hat{\beta}$

end for

In the final cycle, for each j solve for $\hat{\theta}_{12} = \hat{\beta}.\hat{\theta}_{22}$ with $1/\hat{\theta}_{22} = s_{22} - \omega_{12}^{\top}\hat{\beta}$; finally set $\omega_{22} = s_{22}$ since the diagonal of Γ is zero.

Estimation of the graph structure

- Very often, we don't know the structure of the graph and need to discover it from the data.
- Meinshausen and Bühlmann (2006) proposed a simple lasso method to estimate the nonzero elements of $\Theta = \Sigma^{-1}$.
- They suggested to treat each variable X_j as the response variable and regress it on all other variables, X_{-j}, using lasso regression models.
- Then, the component θ_{jk} is estimated to be nonzero if either the regression coefficient of X_j on X_k is nonzero or (we could also use AND) the regression coefficient of X_k on X_j is nonzero.
- They show that this procedure asymptotically provides a consistent estimate of the set of non-zero elements of Θ.

Estimation of the graph structure

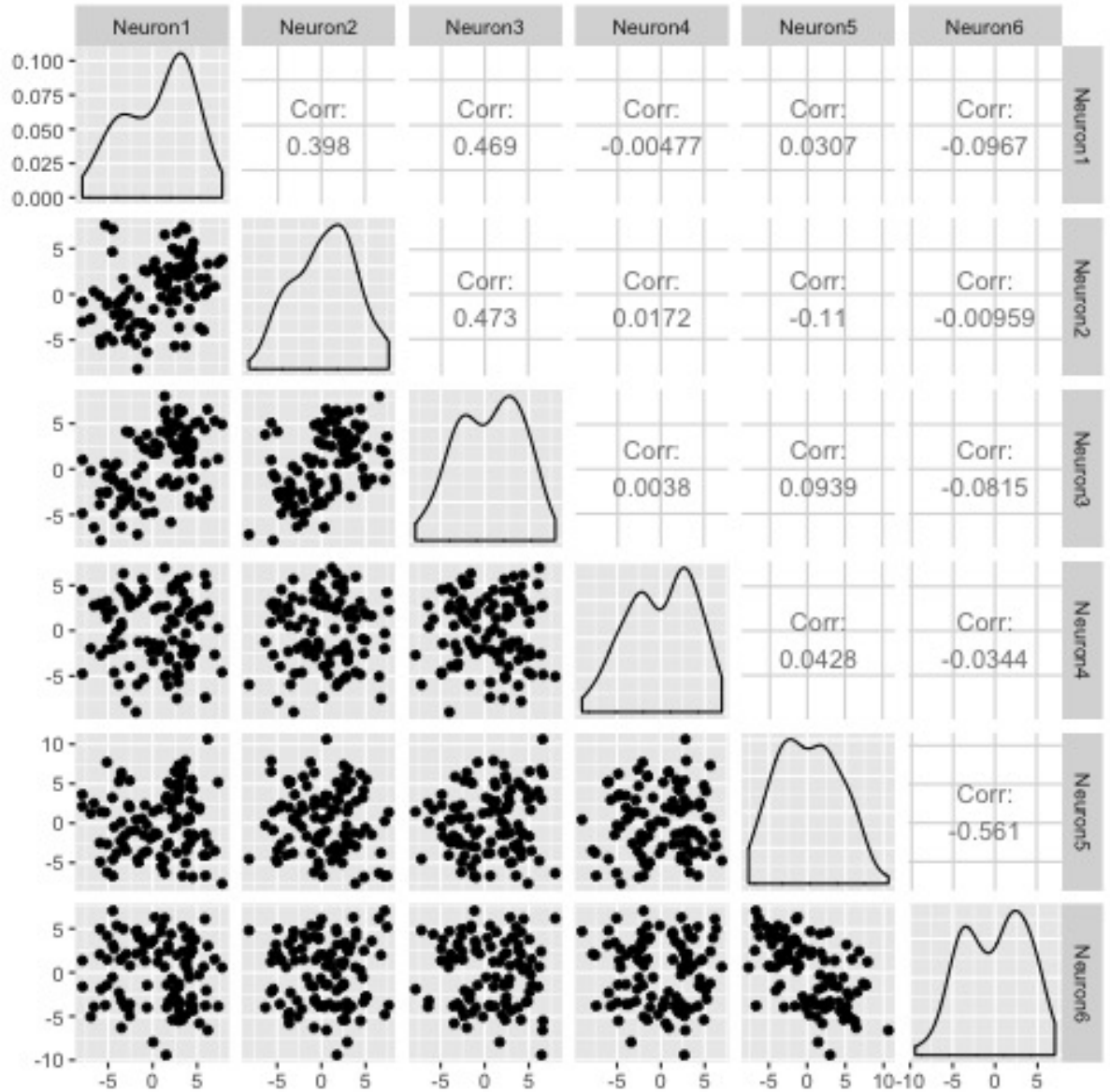
- Alternatively, we could use the lasso penalty to maximized the following penalized log-likelihood (up to a constant):

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

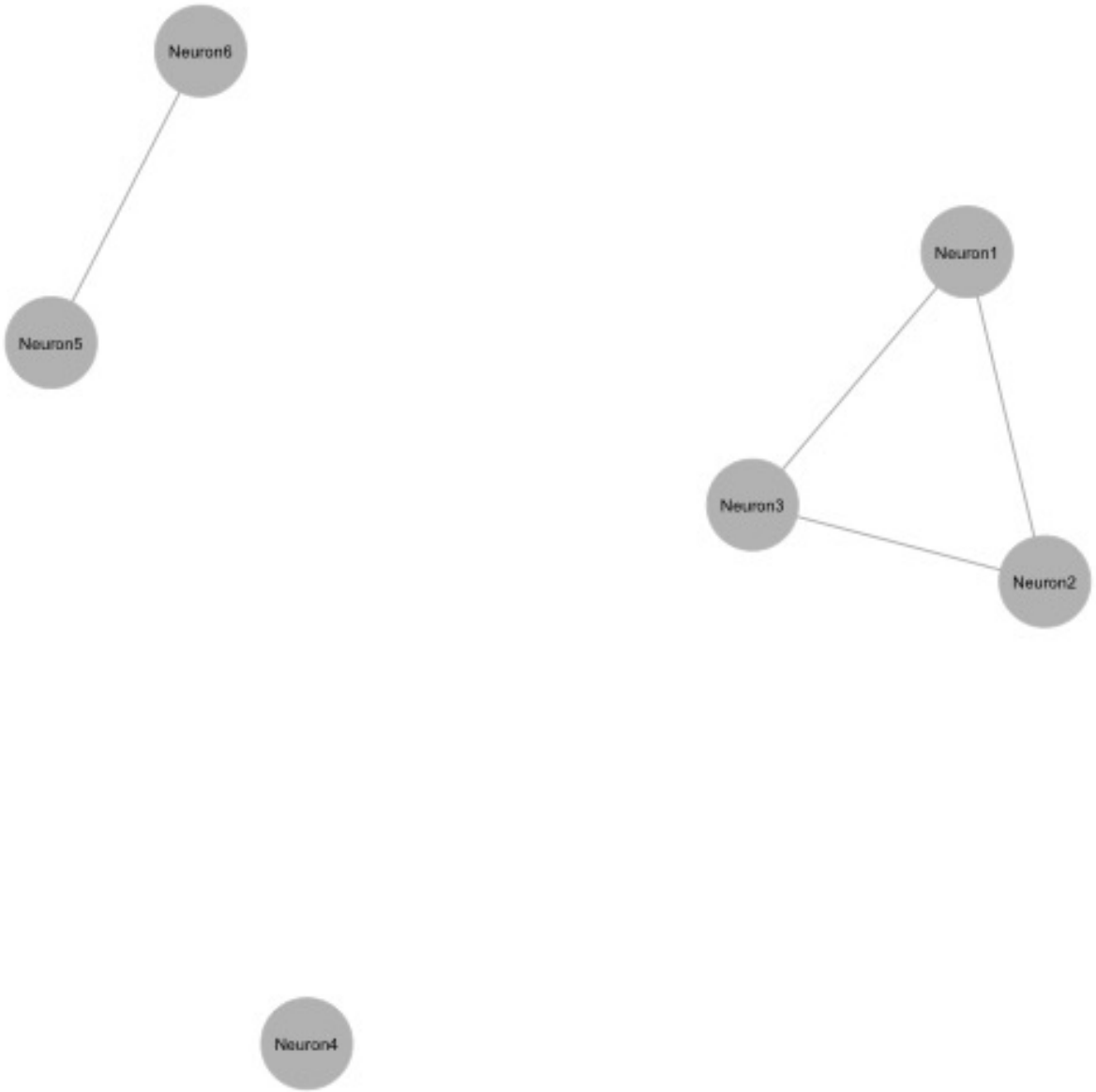
- This approach would lead to a modified version of the previous algorithm.

- It is known as *graphical lasso*, and is implemented in the package `glasso` in R.

Neural coding–Continuous



Neural coding–Continuous



Discrete variables

Ising model

- Recall the Ising model, which includes an array of spins (denoted as ± 1) that are magnetically coupled to each other, with the energy of the a specific configuration, X, given by Hamiltonian function,

$$E(X, \alpha, \beta) = -\frac{1}{2} \sum_{i,j} \beta_{ij} x_i x_j - \sum_i \alpha_i x_i$$

- β_{ij} represents the coupling (interaction) between two neighboring spins such that $\beta_{ij} = 0$ if $(i, j) \notin \mathcal{N}$.

- α_i represents the external magnetic field on spin i.

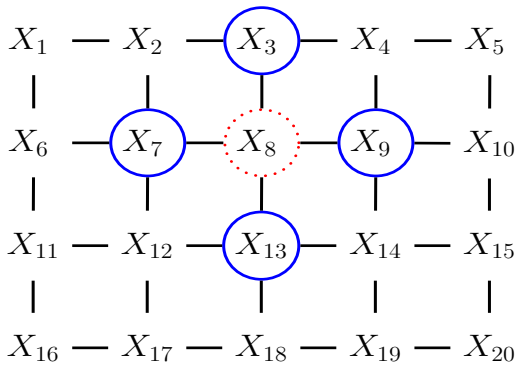


Fig19.1b in Murphy (2012)

Ising model

- The probability of any specific configuration, X, is given by the Boltzmann distribution,

$$\begin{aligned} P(X|\alpha,\beta,T) &= \frac{1}{z(\alpha,\beta,T)}\exp\{-\frac{1}{K_B T}E(X,\alpha,\beta)\} \\ &= \frac{1}{z(\alpha,\beta,T)}\exp\{\frac{1}{K_B T}[\frac{1}{2}\sum_{i,j}\beta_{ij}x_i x_j + \sum_i \alpha_i x_i]\} \end{aligned}$$

where,

$$\begin{aligned} z(\alpha,\beta,T) &= \sum_x \exp\{-\frac{1}{K_B T}E(X,\alpha,\beta)\} \\ P(X_i = +1|.) &= \frac{1}{1+\exp(-\frac{2}{K_B T}[\sum_j \beta_{ij}x_j + \alpha_i])} \end{aligned}$$

- Here, T is the temperature and K_B is Boltzmann's constant.

Boltzmann machines

- In statistical machine learning, the Ising model is use to represent the joint distribution of a set of binary variables (nodes, units) structured on a graph.

- Here, we assume $X \in \{0,1\}^p$ (i.e., instead of ± 1 commonly used in statistical mechanics), and write the joint distribution given model parameters as follows:

$$P(X|\Theta) = \exp[\sum_{(j,k)\in E} \theta_{jk} X_j X_k - \Phi(\Theta)]$$

  where $\Phi(\Theta)$ is the log of the partition function.

- Note that in this parameterization, instead of the external magnetic field, we assume that there is a *constant* node $X_0 = 1$, which is connected to all other nodes.

Boltzmann machines

- The above model is similar to the Poisson log-linear model commonly used for multiway tables in statistics (see Agresti 2002 for example).
- In statistical machine learning, we refer to it as *Boltzmann machine*.
- The conditional probability in this case is

$$P(X_j = 1|.) \quad = \quad \frac{1}{1 + \exp(-\theta_{j0} - \sum_{(j,k)\in E} \theta_{jk} x_k)}$$

which is a logistic regression model of X_j on all other variables. Therefore, θ_{jk} captures the dependence of X_j on X_k given all other variables.

Parameter estimation given the network structure

- When the structure of the graph is given, we can estimate the model parameters by maximizing the log-likelihood function

$$\ell(\Theta) = \sum_{i=1}^{n} \log P(X_i = x_i|\Theta) = \sum_{i=1}^{n}[\sum_{(j,k)\in E} \theta_{jk} x_{ij} x_{ik} - \Phi(\Theta)]$$

- The gradient function is

$$\frac{\partial \ell(\Theta)}{\partial \theta_{jk}} = \sum_{i=1}^{n} x_{ij} x_{ik} - n \frac{\partial \Phi(\Theta)}{\partial \theta_{jk}}$$

where

$$\frac{\partial \Phi(\Theta)}{\partial \theta_{jk}} = \sum_{x} x_j x_k p(x|\Theta) = E_\Theta(X_j X_k)$$

- Setting the gradient to zero, we obtain the following score function:

$$\frac{1}{n} \sum_{i=1}^{n} x_{ij} x_{ik} = E_\Theta(X_j X_k)$$

Parameter estimation given the network structure

- Note that this is consistent with what we mentioned earlier about exponential family models: the score function sets the sufficient statistic to its theoretical expectation under the model.
- For small scale problems, we can use simple gradient-based optimization methods (e.g., Newton) to estimate the parameters.
- For larger problems, we can use approximation methods (e.g., mean field).
- Sometimes, our graph structure includes *hidden* nodes, which represent *latent* (i.e., unobservable) variables.
- In this case, we write down the likelihood function by summing over all hidden nodes (i.e., all possible $\{0, 1\}$ combinations for the hidden nodes) similar to what we used for mixture models with latent mixture indicators.

Resticted Boltzmann Machine (RBM)

- The hidden and visible nodes can be connected to each other without any restrictions, such as the graph shown in left panel.
- However, sometimes for simplicity, we arrange the hidden nodes and visible nodes in different layers such that there are no connections between nodes in the same layer. We refer to the resulting model as *Restricted Boltzmann Machine (RBM)*.
- The figure in the right panel shows an example of RBM.

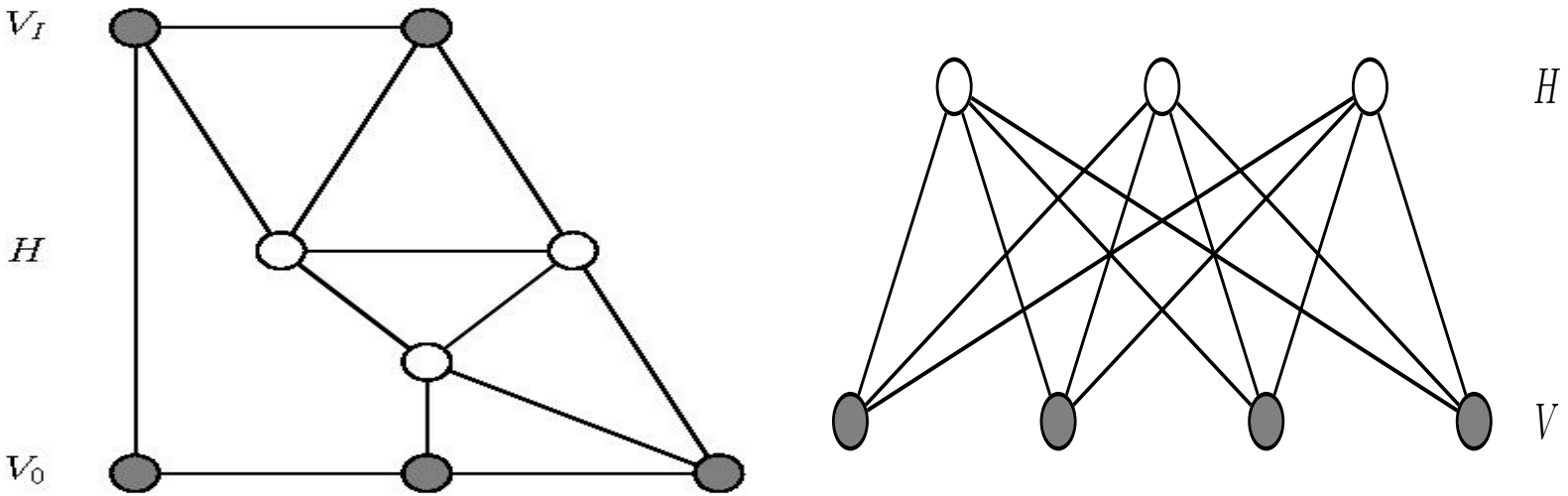


Fig27.30 in Murphy (2012)

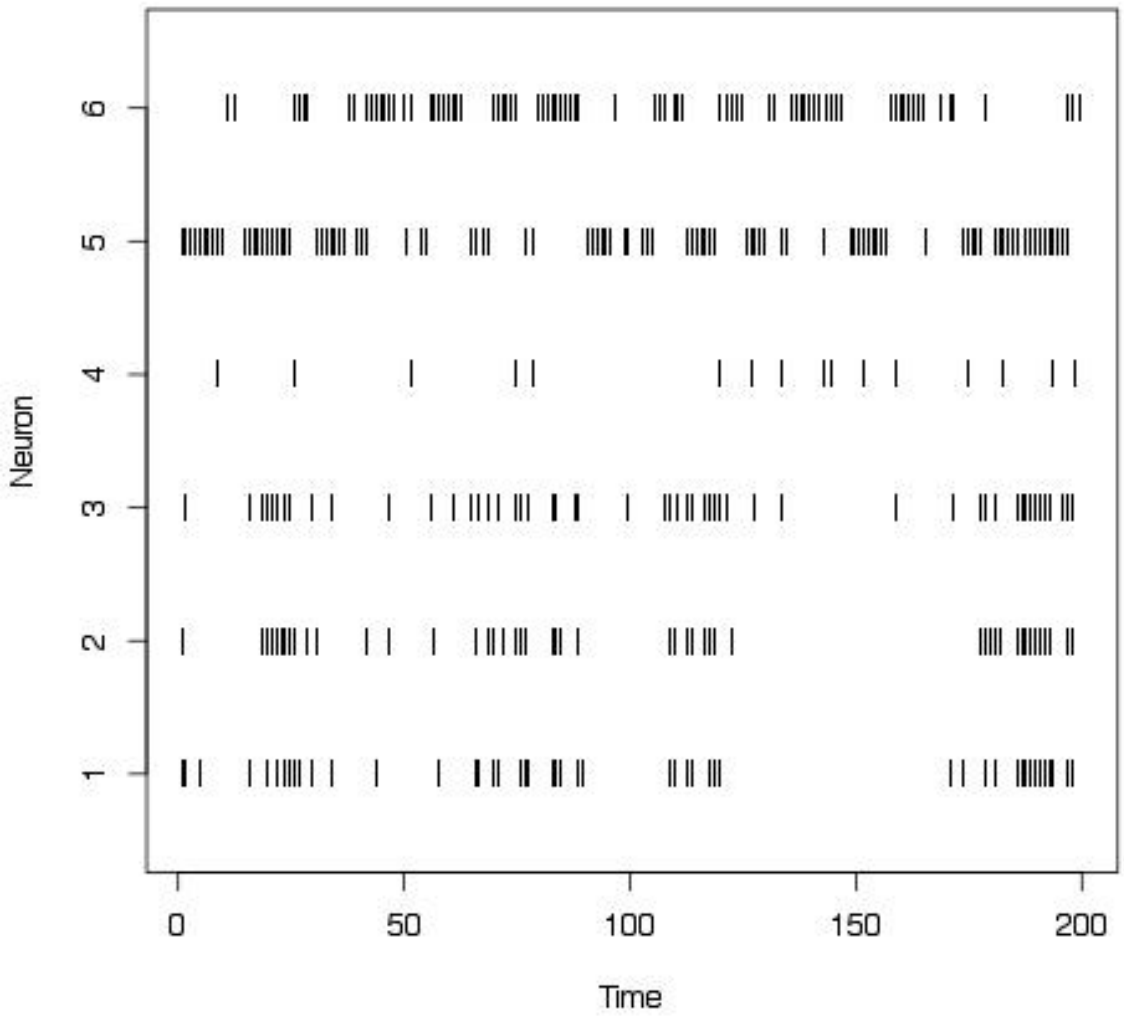
Estimation of the topology of a graph

- Similar to undirected graphical models with continuous variables, we rarely know the graph structure; we need to estimate it using the observed data.
- As before, we can use the lasso penalty to enforce sparsity (i.e., remove some edges).
- Wainwright et. al. (2007) proposed a method analogous to the Meinshausen and Bühlmann (2006) approach, but it uses L_1-penalized logistic regression models instead.
- After estimating all the parameters, $\tilde{\theta}_{jk}$, we can force symmetry by using, for example,

$$\hat{\theta}_{jk} = \hat{\theta}_{kj} = \min(|\tilde{\theta}_{jk}|, |\tilde{\theta}_{kj}|)$$

  or we could use the "max" symmetrization in a similar way.
- Alternatively, we could use the extended graphical lasso model for discrete Markov networks proposed by Hoefling and Tibshirani (2008).

Neural coding–Discrete



Neural coding–Discrete