Undirected Graphical Models

1.Introduction and Markov Graphs
2.UGM for Continuous Variables → Jianfeng Liang

3.UGM for Discrete Variables → Zhe Xu

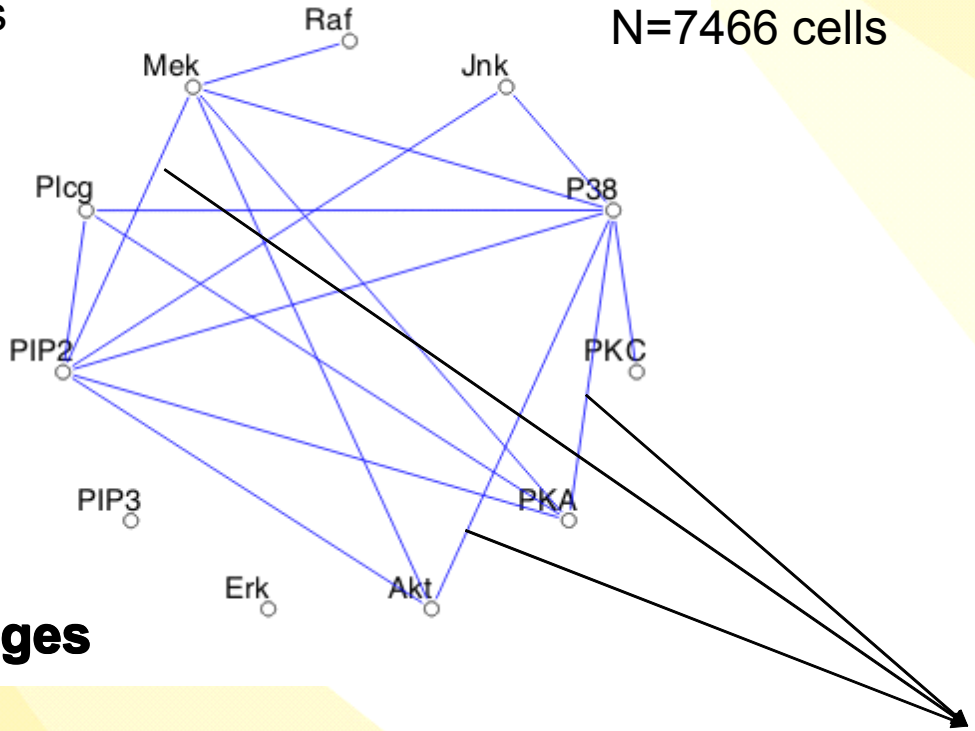
1.Introduction and Markov Graphs

Figure 17.1

a graphical model for a flow-cytometry dataset

p=11 proteins

N=7466 cells



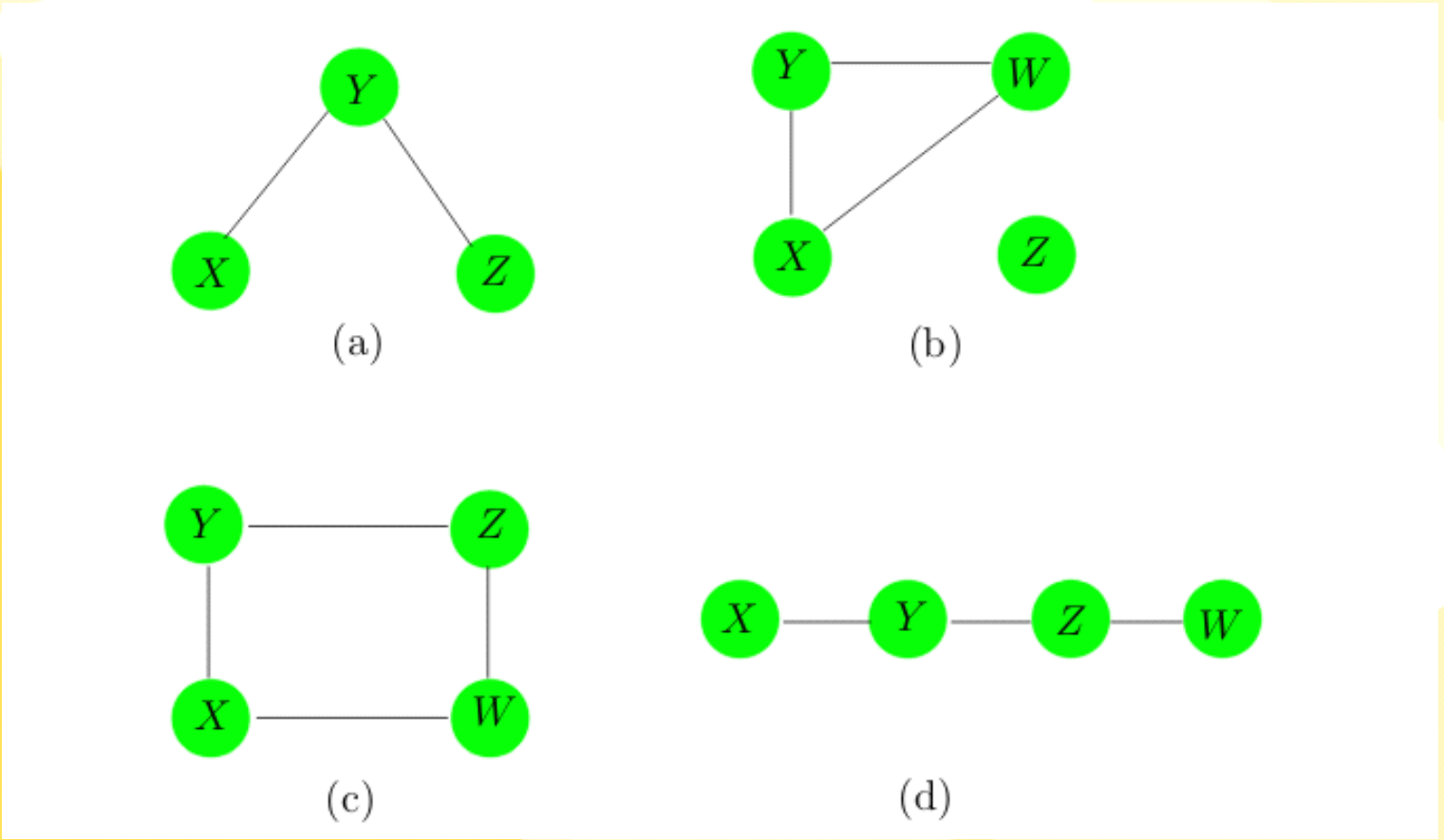
main challenges

1.model selection;

2.estimation (*learning*);

3.computation of marginal vertex probilities & expectations.(*inference*)

- **No edge joining X and Y** $X \perp Y \mid \text{rest}$
- **if C separates A and B then** $A \perp B \mid C$.



$$f(x) = 1/Z \prod_{C \in \mathsf{C}} \psi_C(x_C)$$

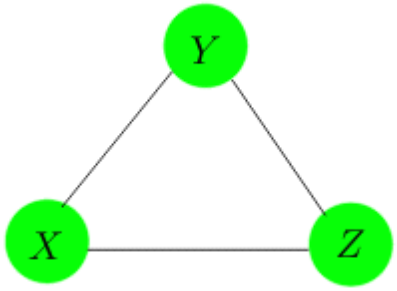
$$Z = \sum_{x \in \mathsf{X}} \prod_{C \in \mathsf{C}} \psi_C(x_C),$$

C is the set of maximal cliques, and the positive functions $\psi_C(\bullet)$ are called clique potentials.

These are afinities that capture the dependence in X_C by scoring certain instance x_C higher than others.

Z is the partition function.

The representation of f(x) implies a graph with independence properties defined by the cliques in the product.



A complete graph does not uniquely specify the higher-order dependence structure in the joint distribution of the variables.

$$f^{(2)}(x,y,z) = 1/Z\psi(x,y)\psi(y,z)\psi(x,z);$$

$$f^{(3)}(x,y,z) = 1/Z\psi(x,y,z).$$

2.UGM for Continuous Variables

$$Y|Z=z \sim N(\mu_Y+(z-\mu_Z)T\sum_{ZZ}^{-1}\sigma_{ZY},\sigma_{YY}-\sigma_{ZY}^{T}\sum_{ZZ}^{-1}\sigma_{ZY}),$$

Where we have partitioned Σ as $\Sigma=\begin{pmatrix}\Sigma_{ZZ} & \sigma_{ZY}\\ \sigma_{ZY}^{T} & \sigma_{YY}\end{pmatrix}$.

According to what we get in the Statistical Decision Theory,

$\beta=\sum_{ZZ}^{-1}\sigma_{ZY}$.

We can partition Θ in the same way. Since $\Sigma\Theta=\mathrm{I}$, we have

$$\theta_{ZY}=-\theta_{YY}\cdot\sum_{ZZ}^{-1}\sigma_{ZY},$$

where $1/\theta_{YY}=\sigma_{YY}-\sigma_{ZY}^{T}\sum_{ZZ}^{-1}\sigma_{ZY}>0.$ Then we find $\beta=\sum_{ZZ}^{-1}\sigma_{ZY}=-\theta_{ZY}/\theta_{YY}.$

We have got two things :

1. Here zero elements in $\boldsymbol{\beta}$ and hence $\boldsymbol{\theta}_{ZY}$ mean that the corresponding elements of Z are conditionally independent of Y, given the rest.
2. We can learn about this dependence structure through multiple linear regression.

Estimation when the Graph Structure is Known

- 1.With some realization of X, we want to estimate the parameters in a graph that approximates their joint distribution.

- Suppose that we have N multivariate normal realizations x_i, i=1,...,N with population mean μ and covariance Σ in a complete (fully connected) graph.
- Let

$$S = (1/N) \cdot \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T$$

- be the empirical covariance matrix, with $\bar{x}$ the sample mean vector.
- Ignoring constants, the log-likelihood of the data can be written as
- $\ell(\Theta) = \log\det\Theta - trace(S\Theta)$ maximized partially with respect to μ.
- According to $\ell(\Theta)$, we have the MLE of Σ is S.

- 2.To constrain the log-likelihood above for all missing edges, we add Lagrange constants

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

- So we will get the maximizing gradient equation as

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

- Here, Θ^{-1} equals the derivate of log det Θ, and Γ is a matrix of Lagrange parameters with nonzero values for all pairs with edges absent.
- Next, use regression to solve for Θ and its inverse $W=\Theta^{-1}$ one row and column at a time.

$$w_{12} - s_{12} - \gamma_{12} = 0.$$

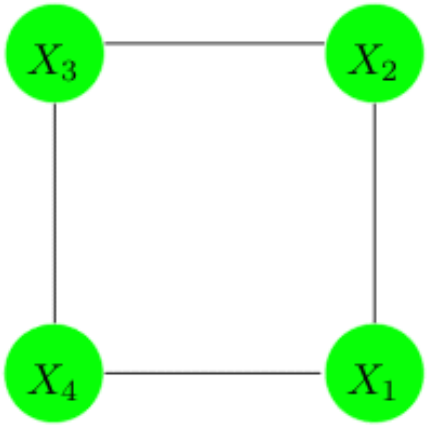
- As above we did, partition the matrices into two parts. We have
- $\begin{pmatrix} W_{11} & w_{12} \\ w_{11}^T & w_{22} \end{pmatrix}\begin{pmatrix} \Theta_{11} & \theta_{12} \\ \theta_{12}^T & \theta_{22} \end{pmatrix}=\begin{pmatrix} \mathrm{I} & 0 \\ 0^T & 1 \end{pmatrix}$. It implies $w_{12}=-W_{11}\theta_{12}/\theta_{22}=W_{11}\beta$
- where $\beta=-\theta_{12}/\theta_{22}$.
- Now substituting w_{12} gives $W_{11}\beta-s_{12}-\gamma_{12}=0$.
- Finally, Suppose there are p−q nonzero elements in γ_{12}—i.e., p−q edges constrained to be zero.

$$W_{11}^*\beta^*-s_{12}^*=0\Rightarrow\hat{\beta}^*=W_{11}^{*-1}s_{12}^*.$$

- By using partitioned inverse formulas, we have $1/\theta_{22}=w_{22}-w_{12}^T\beta$,
- where $w_{22}=s_{22}\Leftarrow\Gamma=0.$

- So, we have learned three steps for the estimation. They are
- 1. Initialize W = S.
- 2. repeat for j = 1,2,...,p until convergence:
- (a) Partition the matrix W into part 1: all but the jth row and
- column, and part 2: the jth row and column.
- (b) Solve $W_{11}^*\beta^*-s_{12}^*=0$ for the unconstrained edge parameters β^*. Obtain $\hat{\beta}$ by padding $\hat{\beta}^*$with zeros in the appropriate positions.

- (c) Update $w_{12} = W_{11}\hat{\beta}$.
- 3. In the final cycle (for each j) solve for $\hat{\theta}_{12} = -\hat{\beta} \cdot \hat{\theta}_{22}$, with $1/\theta_{22} = w_{22} - w_{12}^T \beta$.



$$\mathbf{S} = \begin{pmatrix} 10 & 1 & 5 & 4 \\ 1 & 10 & 2 & 6 \\ 5 & 2 & 10 & 3 \\ 4 & 6 & 3 & 10 \end{pmatrix}$$

- Taking a little example, we apply our method to the problem above; in the modified regression for variable 1 in step (b), variable 3 is left out. The procedure quickly converged to the solutions:

$$\hat{\mathbf{\Sigma}} = \begin{pmatrix} 10.00 & 1.00 & 1.31 & 4.00 \\ 1.00 & 10.00 & 2.00 & 0.87 \\ 1.31 & 2.00 & 10.00 & 3.00 \\ 4.00 & 0.87 & 3.00 & 10.00 \end{pmatrix}, \quad \hat{\mathbf{\Sigma}}^{-1} = \begin{pmatrix} 0.12 & -0.01 & 0.00 & -0.05 \\ -0.01 & 0.11 & -0.02 & 0.00 \\ 0.00 & -0.02 & 0.11 & -0.03 \\ -0.05 & 0.00 & -0.03 & 0.13 \end{pmatrix}.$$

Estimation of the Graph Structure

- In most cases we do not know which edges to omit from our graph, and so would like to try to discover this from the data itself.
- A simple approach: rather than trying to fully estimate Σ or $\Theta = \Sigma^{-1}$, we can only estimate which components of θ_{ij} are nonzero.
- A more systematic approach: consider maximizing the penalized log-likelihood $\log\det\Theta - trace(S\Theta) - \lambda\|\Theta\|_1$
- where $\|\Theta\|_1$ is the lasso norm—the sum of the absolute values of the elements of Σ^{-1}, and we have ignored constants.
- Next,

$$\Theta^{-1} - S - \lambda \cdot Sign(\Theta) = 0 \Rightarrow W_{11}\beta - s_{12} + \lambda \cdot Sign(\beta) = 0$$

- The lasso minimizes $1/2\,(y - Z\beta)^T (y - Z\beta) + \lambda \cdot \|\beta\|_1$
- The gradient of this expression is

$$Z^T Z\beta - Z^T y + \lambda \cdot Sign(\beta) = 0.$$

- To solve the modified lasso problem at each stage, we use the pathwise coordinate descent method.
- Letting V=W_{11}, the update has the form

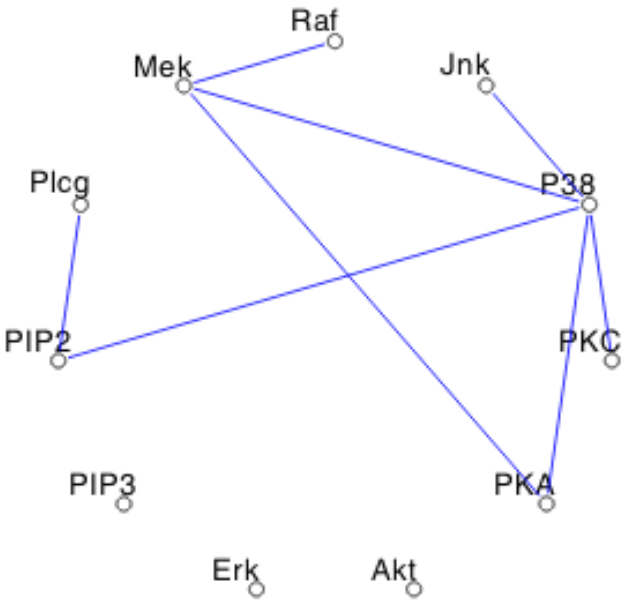
$$\hat{\beta} \leftarrow S(s_{12}j - \sum_{k \neq j} V_{kj}\hat{\beta}_k, \lambda) \Big/ V_{jj}$$

- for j=1, 2,..., p-1, 1, 2,..., p-1,..., where S is the soft-threshold operator: $S(x,t) = sign(x)(|x| - t)_+$.

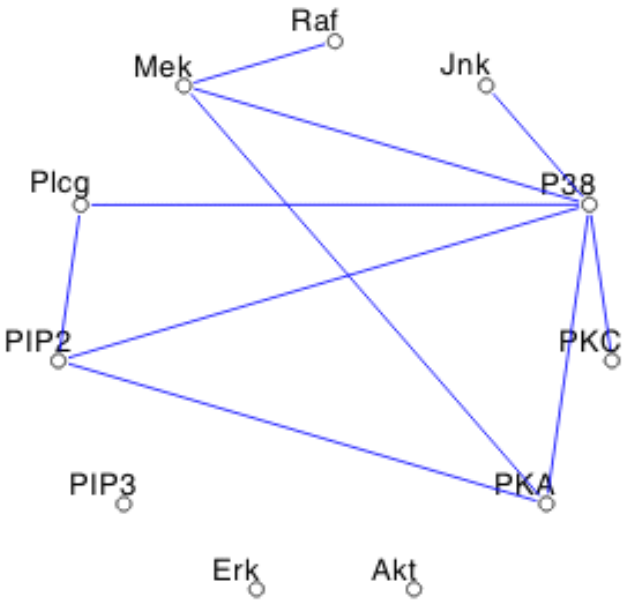
- *Algorithm 17.2 Graphical Lasso.*
- 1. Initialize W = S + λI. The diagonal of W remains unchanged in
- what follows.
- 2. Repeat for j = 1,2,...p,1,2,...p,... until convergence:
- (a) Partition the matrix W into part 1: all but the jth row and
- column, and part 2: the jth row and column.
- (b) Solve the estimating equations $W_{11}\beta$ - s12+ λ • Sign(β) = 0.
- (c) Update $w_{12} = W_{11}\hat{\beta}$.
- 3. In the final cycle (for each j) solve for $\hat{\theta}_{12} = -\hat{\beta} \cdot \hat{\theta}_{22}$, with

$$1/\theta_{22} = w_{22} - w_{12}^T\beta.$$

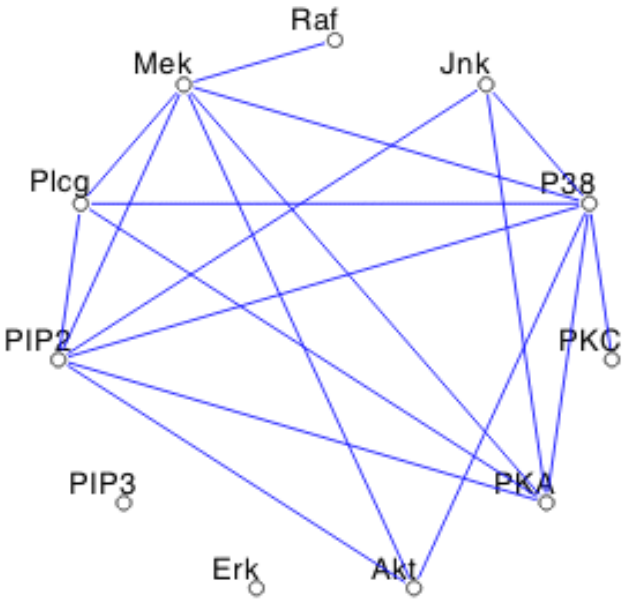
$\lambda = 36$



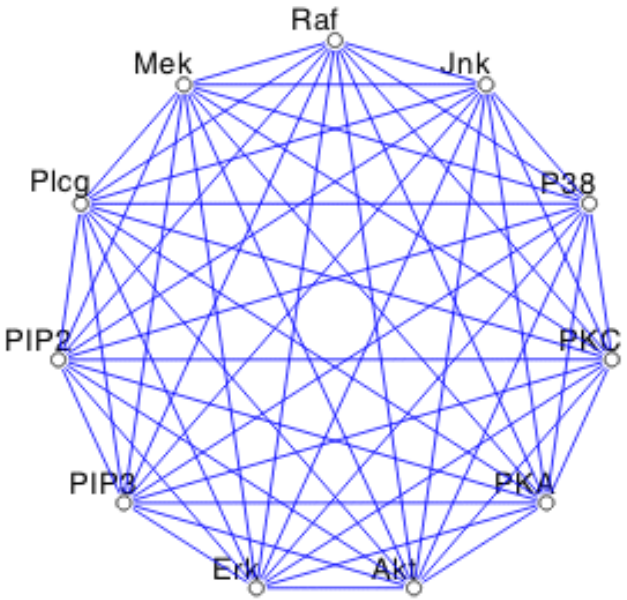
$\lambda = 27$



$\lambda = 7$



$\lambda = 0$



3.UGM for Discrete Variables

- Vertices referred to as "nodes"or "unites"(binary-valued).
- Ising model ~ statistical mechanics literature
  Boltzmann machines ~ machine learning literature
- The values at each node can be observed (“visible”) or unobserved(“hidden”).
- The nodes are often organized in layers.

- We first consider the simpler case in which all nodes are visible with edge pairs (j, k) enumerated in E.(Ising model)
- Joint distribution

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

$$X \in \mathcal{X} = \{0,1\}^p$$

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

$$\Phi(\Theta) = \log \sum_{x\in\mathcal{X}} [\exp(\sum_{(j,k)\in E} \theta_{jk} x_j x_k)]$$

- The parameter θ_{jk} measures the dependence of X_j on X_k, conditional on the other nodes.
- Ising model implies a logistic form for each node conditional on the others:

$$\Pr(X_j = 1 \mid X_{-j} = x_{-j}) = \frac{1}{1+\exp(-\theta_{j0} - \sum_{(j,k)\in E} \theta_{jk} x_k)}$$

- X_{-j} denotes all of the nodes except j.

Estimation when the Graph Structure is Known

- Suppose we have observations $x_i = (x_{i1}, x_{i2}, ..., x_{ip}) \in \{0,1\}^p$, $i = 1, 2..., N$
- The log-liklihood is

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

- The gradient of the log-likelihood is

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

and

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

- Setting the gradient to zero gives $\hat{E}(X_j X_k) - E_{\Theta}(X_j X_k) = 0$

- To find the maximum likelihood estimates, we can use gradient search or Newton methods. However the computation of $E_{\Theta}(X_j X_k)$ is not generally feasible for large p (e.g., larger than about 30).

- For smaller p, a number of standard statistical approaches are available:
  - Poisson log-linear modeling
  - Gradient descent
  - Iterative proportional fitting(IPF)

- When p is large (> 30) other approaches have been used to approximate the gradient:
  - The mean field approximation
  - Gibbs sampling(successively sampling from the estimated model probabilities $\Pr_{\Theta}(X_j | X_{-j})$)

Hidden Nodes

- Suppose that a subset of the variables $\mathcal{X}_{\mathcal{H}}$ are hidden ,and the remainder $\mathcal{X}_{\mathcal{V}}$ are visible.
- The log-liklihood of the observed data is

$$\ell(\Theta) = \sum_{i=1}^{N} \log \Pr_{\Theta}(X_{\mathcal{V}} = x_{i\mathcal{V}})$$

$$= \sum_{i=1}^{N} \left[\log \sum_{x_{\mathcal{H}} \in X_{\mathcal{H}}} \exp \sum_{(j,k) \in E} (\theta_{jk} x_{ij} x_{ik} - \Phi(\Theta)) \right]$$

- The gradient works out to be

$$\frac{d\ell(\Theta)}{d\theta_{jk}} = \hat{E}_{\mathcal{V}} E_{\Theta}(X_j X_k | X_{\mathcal{V}}) - E_{\Theta}(X_j X_k)$$

- The first term is an empirical average of $X_j X_k$ if both are visible; if one or both are hidden, they are first imputed given the visible data, and then averaged over the hidden variables.
- The second term is the unconditional expectation of $X_j X_k$.

- The inner expectation in the first term can be evaluated using basic rules of conditional expectation and properties of Bernoulli random variables.

$$E_\Theta(X_j X_k \,|\, X_\mathcal{V} = x_{i\mathcal{V}}) = \begin{cases} x_{ij} x_{ik} & \text{if } j,k \in \mathcal{V} \\ x_{ij} \Pr_\Theta(X_k = 1 \,|\, X_\mathcal{V} = x_{i\mathcal{V}}) & \text{if } j \in \mathcal{V}, k \in \mathcal{H} \\ \Pr_\Theta(X_j = 1, X_k = 1 \,|\, X_\mathcal{V} = x_{i\mathcal{V}}) & \text{if } j,k \in \mathcal{H} \end{cases}$$

- Now two separate runs of Gibbs sampling are required;
  to estimate $E_\Theta(X_j X_k)$ and $E_\Theta(X_j X_k \,|\, X_\mathcal{V} = x_{i\mathcal{V}})$
- In this latter run, the visible units are fixed (“clamped”) at their observed values and only the hidden variables are sampled.

Estimation of the Graph Structure

- The use of a lasso penalty with binary pairwise Markov networks has been suggested by Lee et al. (2007) and Wainwright et al. (2007).

- Conjugate gradient procedure for exact maximization of a penalized log-likelihood(the computation of $E_{\Theta}(X_j X_k)$)

- An approximate solution approach for the Gaussian graphical model
  • an L1-penalized logistic regression model to each node as a function of the other nodes, and then symmetrize the edge parameter estimates.
  • under certain conditions either approximation estimates the nonzero edges correctly as the sample size goes to infinity
  (can handle denser graphs without the computation of $E_{\Theta}(X_j X_k)$)

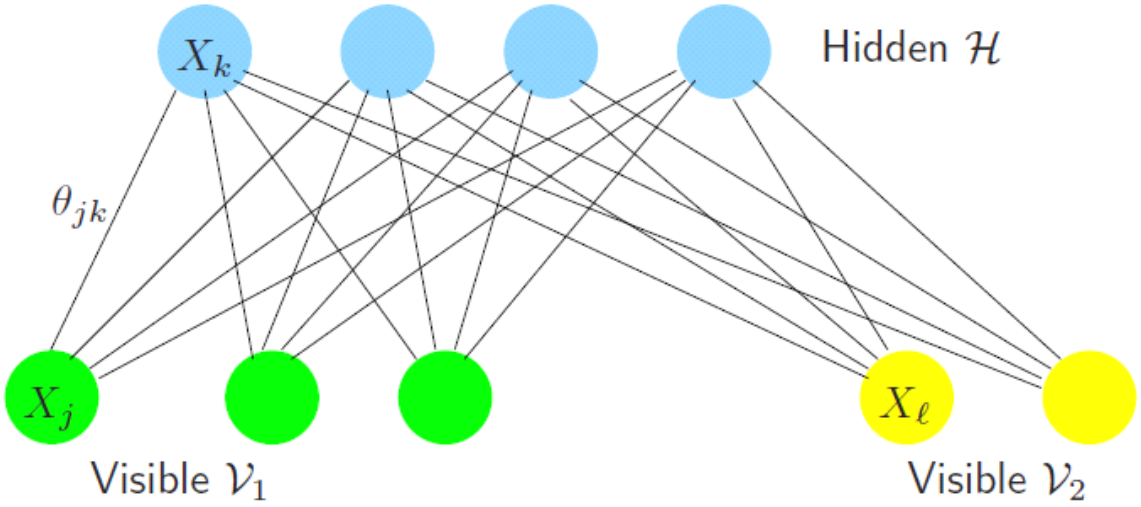
- key difference between the Gaussian and binary models

  • In the Gaussian case, both Σ and its inverse will be of interest, the graphical lasso procedure delivers estimates for both.

  • In the Markov model for binary data, Θ is the object of interest, and its inverse is not of interest.

Restricted Boltzmann Machines

- A restricted Boltzmann machine (RBM) consists of one layer of visible units and one layer of hidden units with no connections within each layer.



- the visible layer is divided into input variables $\mathcal{V}_1$ and output variables $\mathcal{V}_2$, and there is a hidden layer $\mathcal{H}$.

$$\mathcal{V}_1 \leftrightarrow \mathcal{H} \leftrightarrow \mathcal{V}_2$$

- The restricted form of this model simplifies the Gibbs sampling for estimating the expectations in

$$\frac{d\ell(\Theta)}{d\theta_{jk}} = \hat{E}_V E_\Theta(X_j X_k | X_V) - E_\Theta(X_j X_k)$$

since the variables in each layer are independent of one another, given the variables in the other layers.Hence they can be sampled together, using the conditional probabilities given by expression

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

- Restricted Boltzmann machine has the same generic form as a single hidden layer neural network.
  - The neural network minimizes the error
  - The restricted Boltzmann machine maximizes the log-likelihood

- Gibbs sampling in a restricted Boltzmann machine can be very slow.

- Contrastive Divergence:
  estimate $E_\Theta(X_j X_k)$ by starting the Markov chain at the data and only running for a few steps (instead of to convergence)
  sample $\mathcal{H}$ given $\mathcal{V}_1, \mathcal{V}_2$, then $\mathcal{V}_1, \mathcal{V}_2$ given $\mathcal{H}$ and finally $\mathcal{H}$ given $\mathcal{V}_1, \mathcal{V}_2$

- The idea is that when the parameters are far from the solution, it may be wasteful to iterate the Gibbs sampler to stationarity, as just a single iteration will reveal a good direction for moving the estimates.

Example:The MNIST database of handwritten digits

- First, an RBM with 784 visible units and 500 hidden units is trained, using contrastive divergence, to model the set of images.
- Then the hidden states of the first RBM are used as data for training a second RBM that has 500 visible units and 500 hidden units.
- Finally, the hidden states of the second RBM are used as the features for training an RBM with 2000 hidden units as a joint density model.

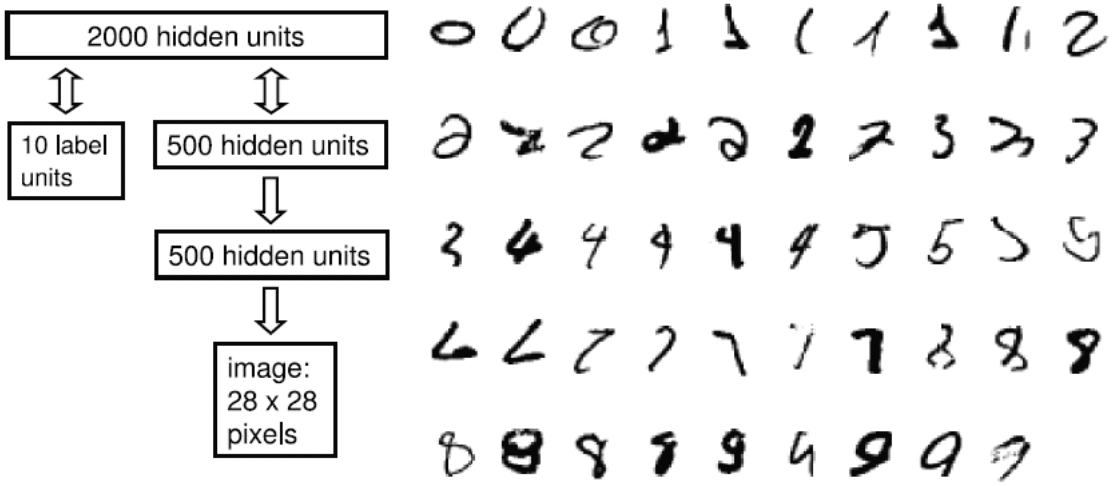


FIGURE 17.7. *Example of a restricted Boltzmann machine for handwritten digit classification. The network is depicted in the schematic on the left. Displayed on the right are some difficult test images that the model classifies correctly.*

THANKS!