GRAPHICAL MODEL STRUCTURE LEAERNING

Presented by Zi Wang

A review of MLPP chapter 26

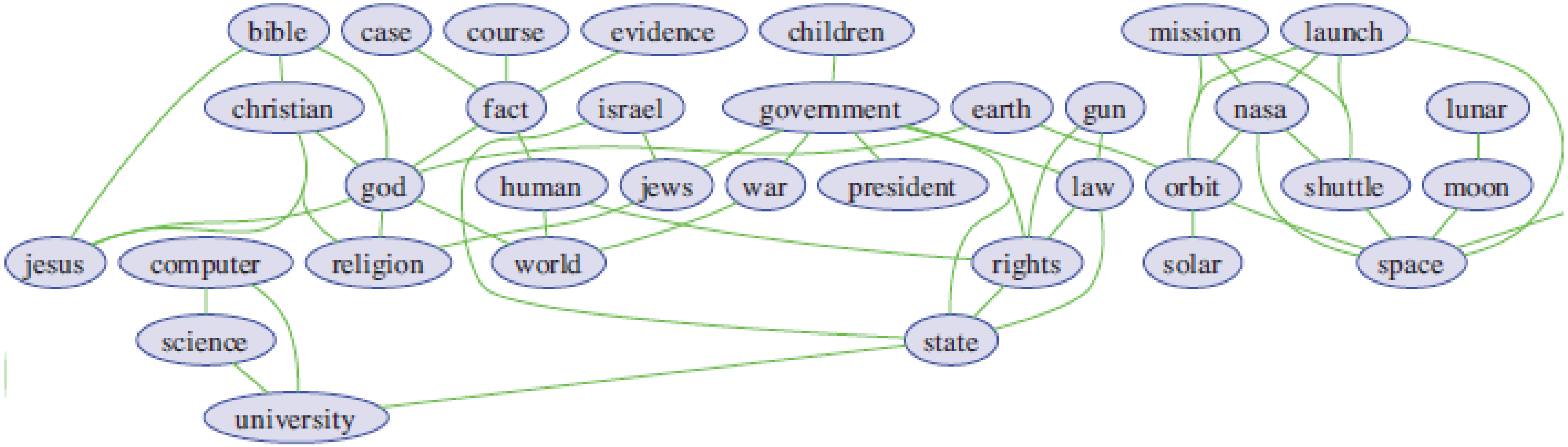
CONTENTS

CONTENTS

- **Structure learning for knowledge discovery**
- Learning tree structures
- Learning DAG structures
- Learning DAG structure with latent variables
- Learning causal DAGs
- Learning undirected Gaussian graphical models
- Learning undirected discrete graphical models

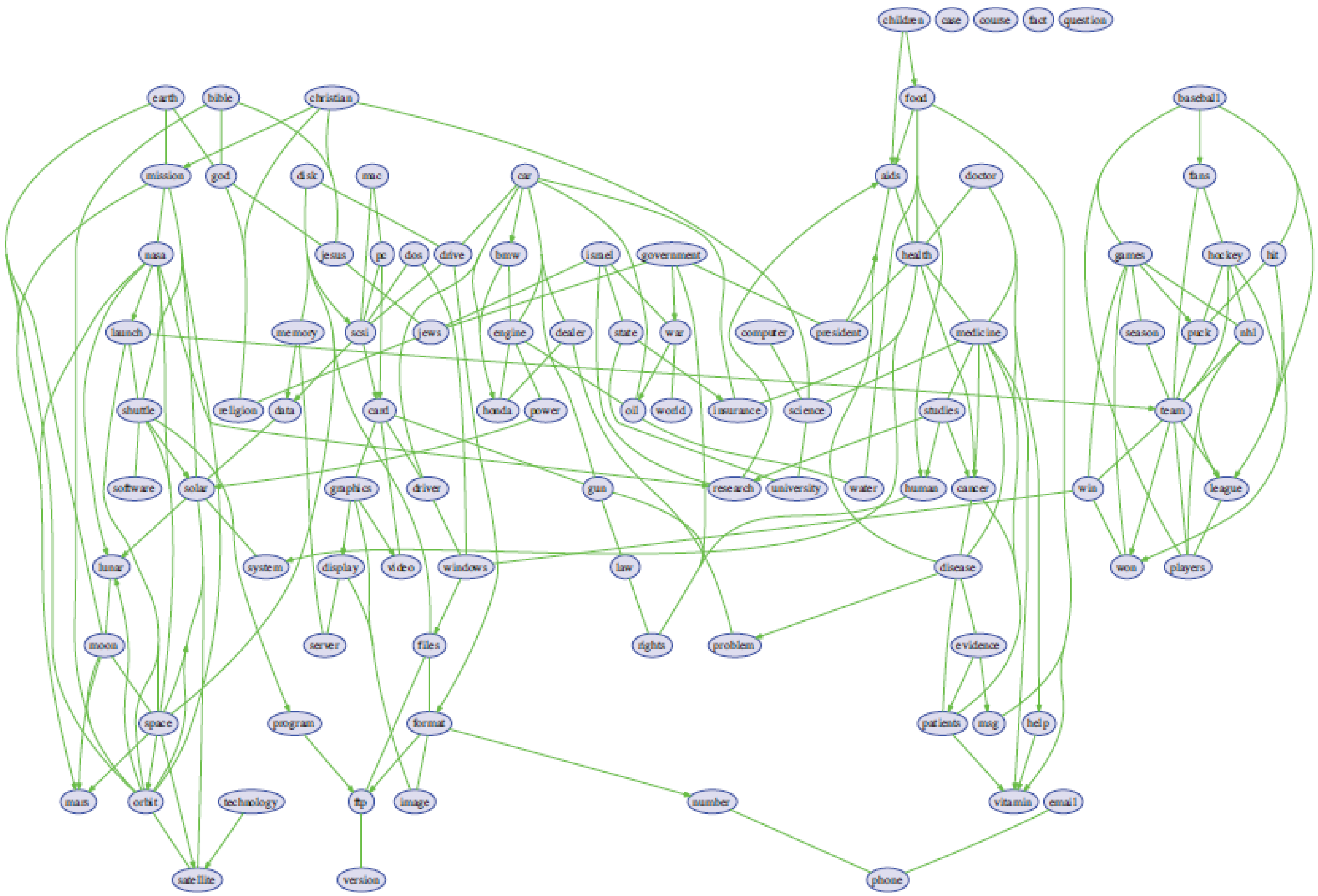
STRUCTURE LEARNING FOR KNOWLEDGE DISCOVERY

- Relevance networks
  - Visualizing mutual information(MI)
  - Dense graph because of disability to capture conditional independence
  - Solution: graphical model



STRUCTURE LEARNING FOR KNOWLEDGE DISCOVERY

- Dependency networks
  - $p(x_t | x_{\neg t})$
  - only its Markov blanket will be chosen as input
  - Sparse graph
- Application
  - Visualization
  - Inference with gibbs sampling



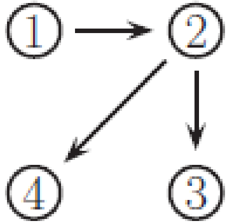
CONTENTS

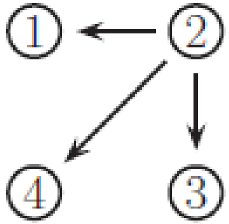
- Structure learning for knowledge discovery
- **Learning tree structures**
- Learning DAG structures
- Learning DAG structure with latent variables
- Learning causal DAGs
- Learning undirected Gaussian graphical models
- Learning undirected discrete graphical models

LEARNING TREE STRUCTURES

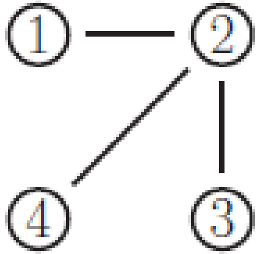
- Equivalence between directed and undirected representation (for trees)

- $p(x_1, x_2, x_3, x_4|T) = p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_2)$
- $p(x_1, x_2, x_3, x_4|T) = p(x_2)p(x_1|x_2)p(x_3|x_2)p(x_4|x_2)$





- $p(x_1, x_2, x_3, x_4|T) = p(x_1)p(x_2)p(x_3)p(x_4)\frac{p(x_1,x_2)}{p(x_1)p(x_2)}\frac{p(x_2,x_3)}{p(x_2)p(x_3)}\frac{p(x_2,x_4)}{p(x_2)p(x_4)}$
- $p(\vec{x}|T) = \prod_{t \in V} p(x_t) \prod_{(s,t) \in E} \frac{p(x_s,x_t)}{p(x_s)p(x_t)}$



CHOW-LIU ALGORITHM FOR FINDING THE ML TREE STRUCTURE

- Log-likelihood function

$$\begin{aligned}\log p(\mathcal{D}|\boldsymbol{\theta}, T) &= \sum_t \sum_k N_{tk} \log p(x_t = k|\boldsymbol{\theta}) \\ &+ \sum_{s,t} \sum_{j,k} N_{stjk} \log \frac{p(x_s = j, x_t = k|\boldsymbol{\theta})}{p(x_s = j|\boldsymbol{\theta}) p(x_t = k|\boldsymbol{\theta})}\end{aligned}$$

CHOW-LIU ALGORITHM FOR FINDING THE ML TREE STRUCTURE

- Log-likelihood function

$$
\begin{aligned}
\frac{\log p(\mathcal{D}|\boldsymbol{\theta}, T)}{N} \quad = \quad & \sum_{t \in \mathcal{V}} \sum_{k} p_{\text{emp}}(x_t = k) \log p_{\text{emp}}(x_t = k) \\
+ & \sum_{(s,t) \in \mathcal{E}(T)} \mathbb{I}(x_s, x_t | \hat{\boldsymbol{\theta}}_{st})
\end{aligned}
$$

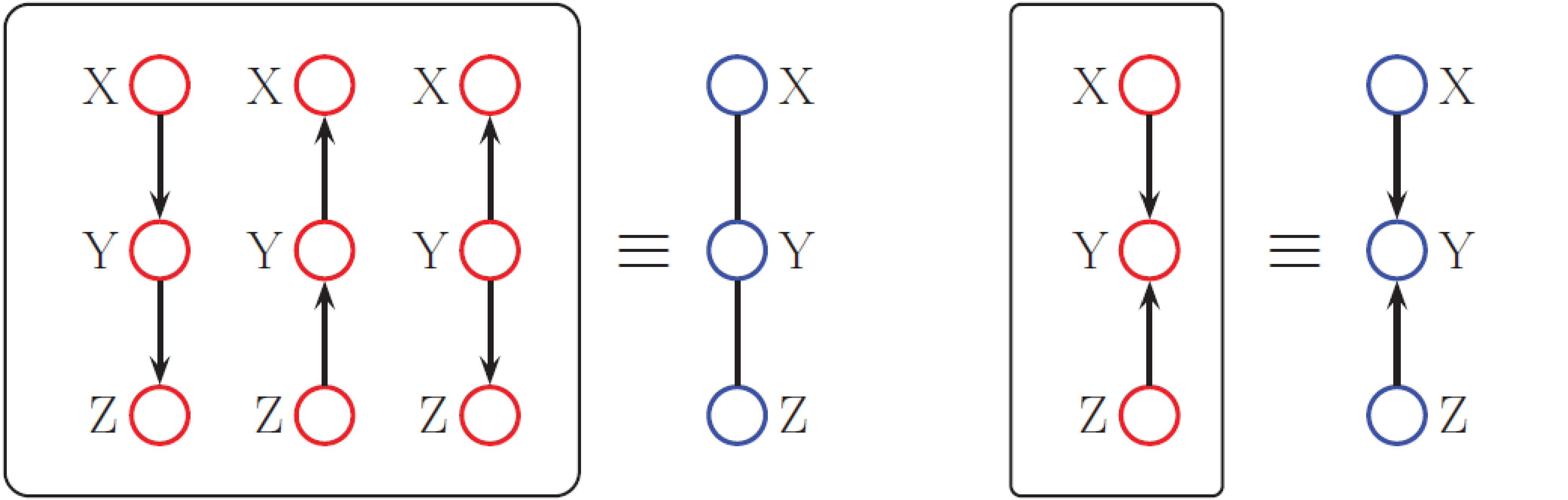
$$
\mathbb{I}(x_s, x_t | \hat{\boldsymbol{\theta}}_{st}) = \sum_{j} \sum_{k} p_{\text{emp}}(x_s = j, x_t = k) \log \frac{p_{\text{emp}}(x_s = j, x_t = k)}{p_{\text{emp}}(x_s = j) p_{\text{emp}}(x_t = k)}
$$

CONTENTS

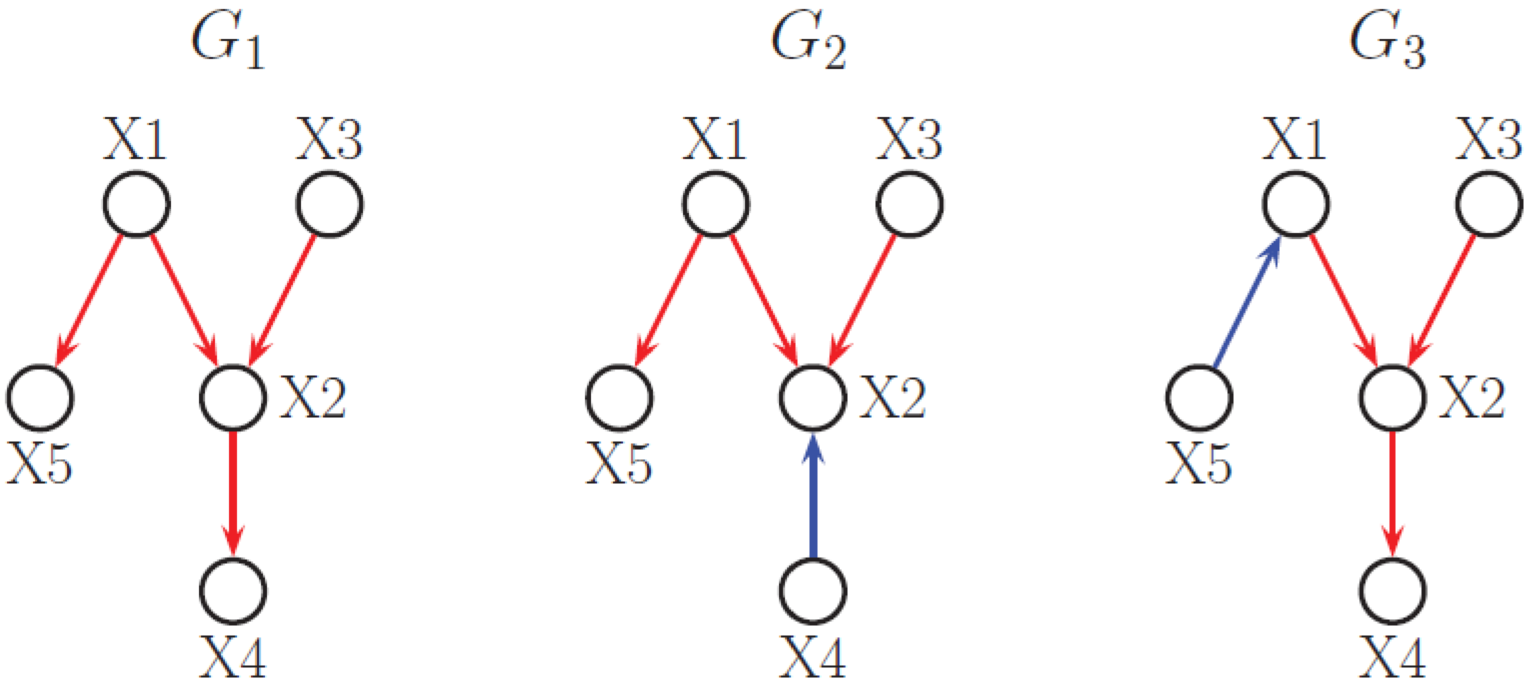
- Structure learning for knowledge discovery
- Learning tree structures
- **Learning DAG structures**
- Learning DAG structure with latent variables
- Learning causal DAGs
- Learning undirected Gaussian graphical models
- Learning undirected discrete graphical models

LEARNING DAG STRUCTURES

- Markov Equivalence



MARKOV EQUIVALENCE - EXAMPLE



LEARNING DAG STRUCTURES

- $\theta_{tck} = p(x_t = k | x_{pa(t)} = c)$ where $c = 1 : C_t,\ C_t = K^{d_t},\ d_t = \dim(pa(t))$
- N_{tck} is the number of times node t is in state k and its parents are in state c
- Assumption: no missing data

$$
\begin{aligned}
p(\mathcal{D}|G, \boldsymbol{\theta}) &= \prod_{i=1}^{N} \prod_{t=1}^{V} \mathrm{Cat}(x_{it}|\mathbf{x}_{i,\mathrm{pa}(t)}, \boldsymbol{\theta}_t) \\
&= \prod_{i=1}^{N} \prod_{t=1}^{V} \prod_{c=1}^{C_t} \mathrm{Cat}(x_{it}|\boldsymbol{\theta}_{tc})^{\mathbb{I}(\mathbf{x}_{i,\mathrm{pa}(t)}=c)} \\
&= \prod_{i=1}^{N} \prod_{t=1}^{V} \prod_{c=1}^{C_t} \prod_{k=1}^{K_t} \theta_{tck}^{\mathbb{I}(x_{i,t}=k, \mathbf{x}_{i,\mathrm{pa}(t)}=c)} \\
&= \prod_{t=1}^{V} \prod_{c=1}^{C_t} \prod_{k=1}^{K_t} \theta_{tck}^{N_{tck}}
\end{aligned}
$$

LEARNING DAG STRUCTURES

- Constraints on priors:
  - $p(\theta) = \prod_t p(\theta_t)$ - global prior parameter independence
  - $p(\theta_t) = \prod_c p(\theta_{tc})$ - local prior parameter independence
  - => $p(\theta_{tc}) = Dir(\theta_{tc}|\alpha_{tc})$

- Marginal likelihood
  - $p(D|G) = \prod_{t=1}^{V} \prod_{c=1}^{C_t} \frac{B(N_{tc}+\alpha_{tc})}{B(\alpha_{tc})}$
  - $score\left(N_t, pa(t)\right) = \prod_{c=1}^{C_t} \frac{B(N_{tc}+\alpha_{tc})}{B(\alpha_{tc})}$
  - Recap Chapter 5:
  - $p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)}$ => $\frac{1}{B(\alpha+N)} \prod_t \theta_t^{\alpha_t+N_t-1} = \frac{\frac{1}{B(\alpha)} \prod_t \theta_t^{\alpha_t-1} \prod_t \theta_t^{N_t}}{p(D)}$ => $p(D) = \frac{B(N+\alpha)}{B(\alpha)}$
  - Here just compute the marginal likelihood directly

LEARNING DAG STRUCTURES

- Setting the prior
- $\alpha_{tck} = \alpha p_0\left(x_t = k, x_{pa(t)} = c\ \right) = \alpha / K_t C_t$
- Example:
- $N_1 = (5,3)\ \ N_{21} = (4,1)\ N_{22} = (1,2)$
- $G_1 = \ X_1 \rightarrow X_2 :\ \alpha_1 = (\frac{\alpha}{2}, \frac{\alpha}{2})\ \alpha_{21} = (\frac{\alpha}{4}, \frac{\alpha}{4})\ \alpha_{22} = (\frac{\alpha}{4}, \frac{\alpha}{4})$
- $G_2 = \ X_1 \quad\ X_2 :\ \alpha_1 = (\frac{\alpha}{2}, \frac{\alpha}{2})\ \alpha_{21} = (\frac{\alpha}{2}, \frac{\alpha}{2})\ \alpha_{22} = (\frac{\alpha}{2}, \frac{\alpha}{2})$

X_1	X_2
1	1
1	2
1	1
2	2
1	1
2	1
1	1
2	2

LEARNING DAG STRUCTURES

- Scaling problem: too many possible graphs

$$f(D) = \sum_{i=1}^{D} (-1)^{i+1} \binom{D}{i} 2^{i(D-i)} f(D-i)$$

- Nodes number: 1 2 3 4 5 6
- Graphs number: 1 3 25 543 29281 3781503
- Solutions
  - Greedy hill climbing
  - Sample DAGs from the posterior, e.g. Metropolis Hasting with proposals in greedy search

CONTENTS

- Structure learning for knowledge discovery
- Learning tree structures
- Learning DAG structures
- **Learning DAG structure with latent variables**
- Learning causal DAGs
- Learning undirected Gaussian graphical models
- Learning undirected discrete graphical models

LEARNING DAG STRUCTURES WITH LATENT VARIABLES

- Learning DAG structure without complete data

$$p(\mathcal{D}|G) = \int \sum_{\mathbf{h}} p(\mathcal{D}, \mathbf{h}|\boldsymbol{\theta}, G)p(\boldsymbol{\theta}|G)d\boldsymbol{\theta} = \sum_{\mathbf{h}} \int p(\mathcal{D}, \mathbf{h}|\boldsymbol{\theta}, G)p(\boldsymbol{\theta}|G)d\boldsymbol{\theta}$$

- Intractable to compute

APPROXIMATING THE MARGINAL LIKELIHOOD WITH MISSING DATA

- BIC approximation

$$\mathrm{BIC}(G) \triangleq \log p(\mathcal{D}|\hat{\boldsymbol{\theta}}, G) - \frac{\log N}{2} \dim(G)$$

- Cheeseman-Stutz approximation

$$p(\mathcal{D}|G) \approx p(\overline{\mathcal{D}}|G) = \int p(\overline{\mathcal{D}}|\boldsymbol{\theta}, G) p(\boldsymbol{\theta}|G) d\boldsymbol{\theta}$$

$$\log p(\mathcal{D}|G) \approx \log p(\overline{\mathcal{D}}|G) + \log p(\mathcal{D}|\hat{\boldsymbol{\theta}}, G) - \log p(\overline{\mathcal{D}}|\hat{\boldsymbol{\theta}}, G)$$

- Variational Bayes EM

$$p(\boldsymbol{\theta}, \mathbf{z}_{1:N}|\mathcal{D}) \approx q(\boldsymbol{\theta}) q(\mathbf{z}) = q(\boldsymbol{\theta}) \prod_i q(\mathbf{z}_i)$$

STRUCTURAL EM

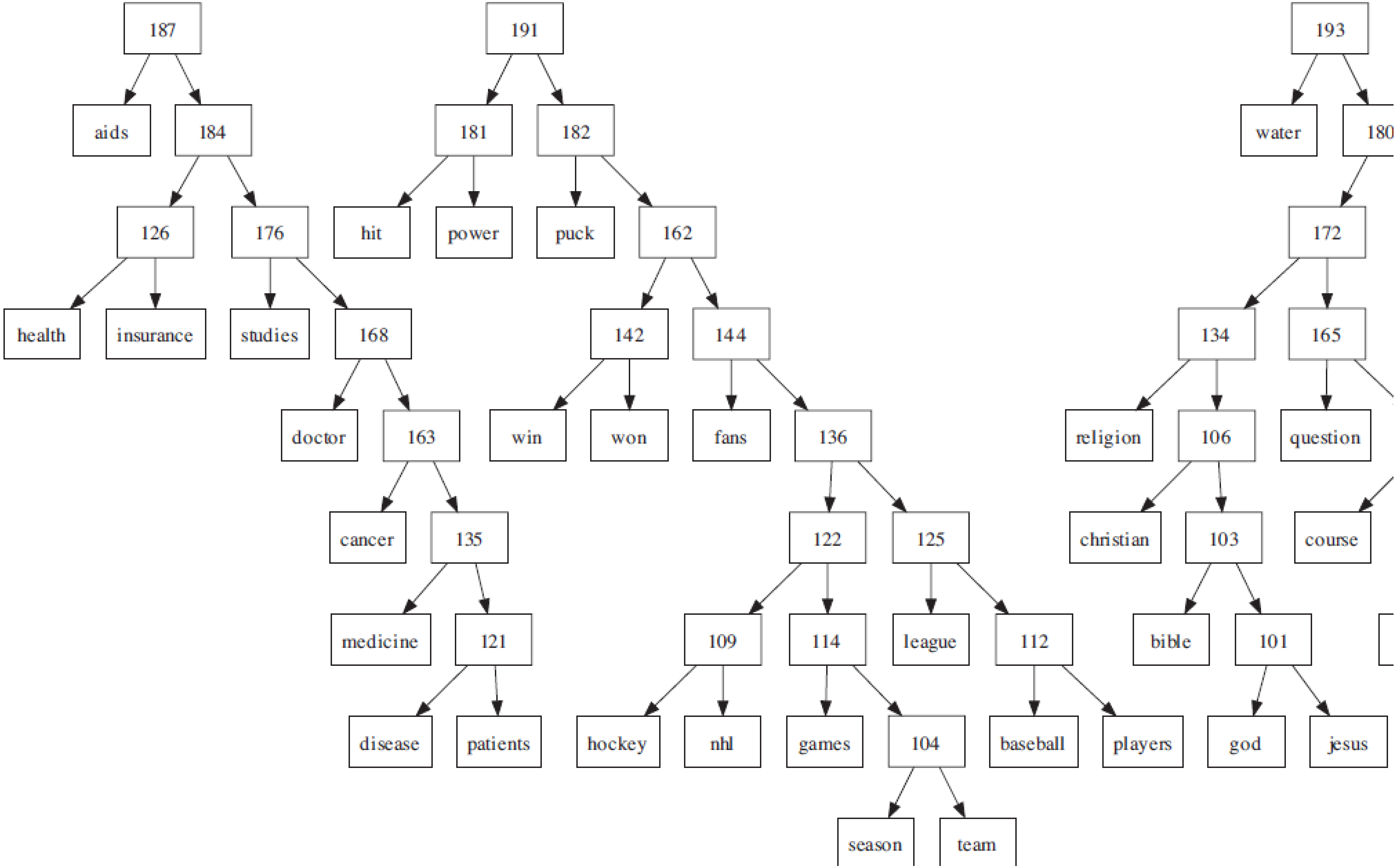
- 0) Initialize a model
- 1) Fill in the data with the current model
- 2) Use the filled-in data to evaluate the score of all the neighbors
- 3) Pick the best neighbor
- 4) Repeat 1),2),3)

$$\text{score}_{\text{BIC}}(G, \mathcal{D}) \triangleq \log p(\mathcal{D}|\hat{\boldsymbol{\theta}}, G) - \frac{\log N}{2}\dim(G) + \log p(G) + \log p(\hat{\boldsymbol{\theta}}|G)$$

- Pro: efficient
- App: learn the phylogenetic tree structure, learn sparse mixture models

DISCOVERING HIDDEN VARIABLES

- Introduce hidden variables to structural signatures, e.g. sets of densely connected nodes
- Latent class model
  - introduce hidden variables with high mutual information with their children
  - Hierarchical latent class model
    - greedy local search algorithm
    - A faster greedy algorithm based on agglomerative hierarchical clustering (Harmeling and Williams 2011)
    - Another approach: Chow-Liu Tree on observed data + add hidden variable


187
aids
184
126
176
health
insurance
studies
168
doctor
163
cancer
135
medicine
121
disease
patients
191
181
182
hit
power
puck
162
142
144
win
won
fans
136
122
125
109
114
league
112
hockey
nhl
games
104
baseball
players
season
team
193
water
180
172
134
165
religion
106
question
christian
103
course
bible
101
god
jesus

STRUCTURAL EQUATION MODELS

SEM: a statistical technique for testing and estimating causal relations using a combination of statistical data and qualitative causal assumptions.

$$x_i = \mu_i + \sum_{j \neq i} w_{ij} x_j + \epsilon_i \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{\Psi})$$

$$\mathbf{x} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \Rightarrow \mathbf{x} = (\mathbf{I} - \mathbf{W})^{-1}(\boldsymbol{\epsilon} + \boldsymbol{\mu})$$

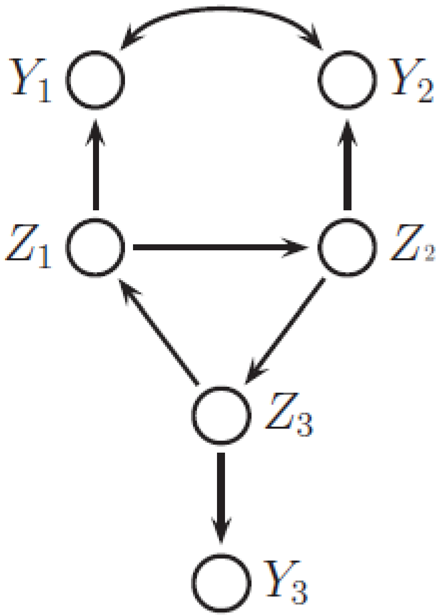
$$p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$$

$$\mathbf{\Sigma} = (\mathbf{I} - \mathbf{W})^{-1}\mathbf{\Psi}(\mathbf{I} - \mathbf{W})^{-T}$$

http://en.wikipedia.org/wiki/Structural_equation_modeling

SEM EXAMPLE

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{pmatrix} = \begin{pmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & w_{13} & 0 & 0 & 0 \\ w_{21} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{32} & 0 & 0 & 0 & 0 \\ w_{41} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{52} & 0 & 0 & 0 & 0 \\ 0 & 0 & w_{63} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{pmatrix}$$



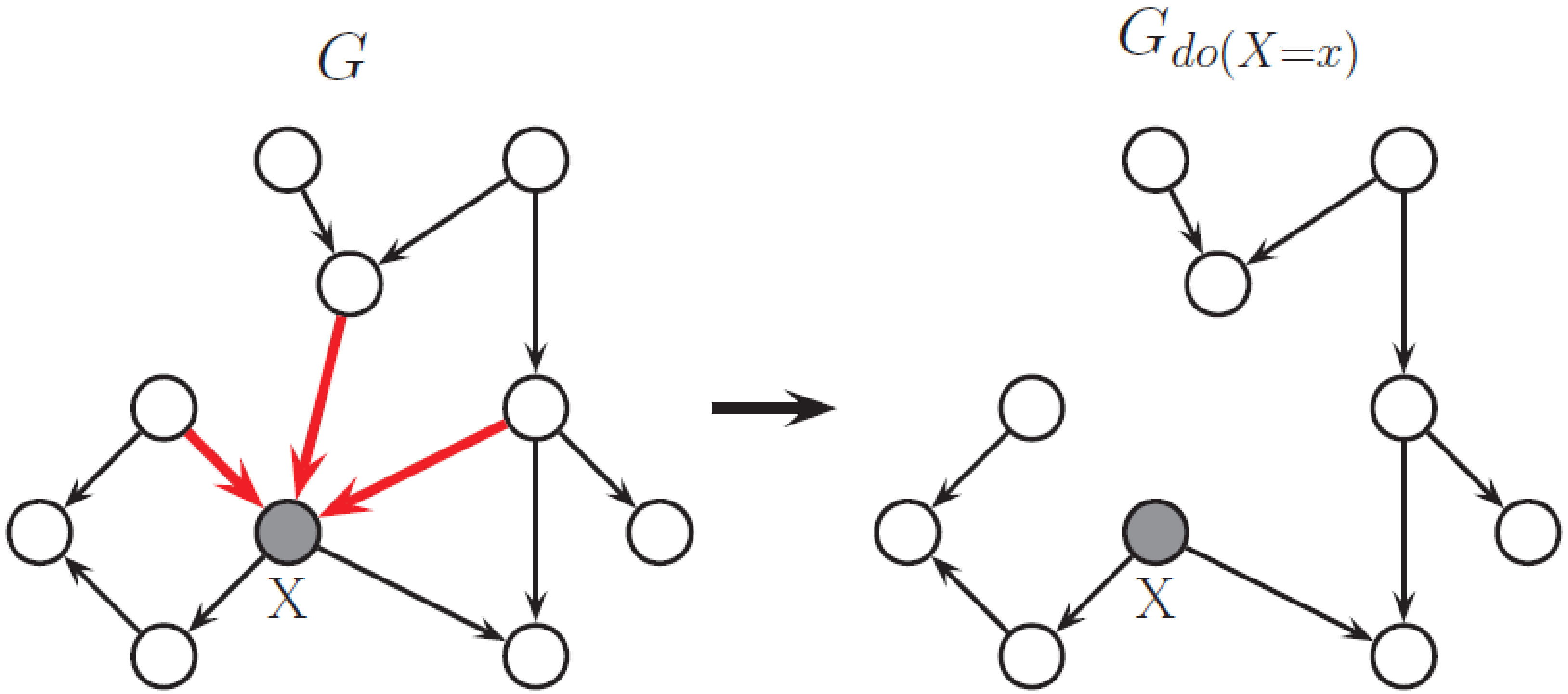
$$\boldsymbol{\Psi} = \begin{pmatrix} \Psi_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & \Psi_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & \Psi_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & \Psi_{44} & \Psi_{45} & 0 \\ 0 & 0 & 0 & \Psi_{54} & \Psi_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & \Psi_{66} \end{pmatrix}$$

CONTENTS

LEARNING CAUSAL DAGS

- Causal models: models which can predict the effects of interventions to, or manipulations of, a system
- Assumptions:
  - Causal Markov assumption
  - Causal sufficiency assumption
- Notations:
  - $do(X_i = x_i)$ – set X_i to be x_i
  - Difference between conditioning on observation and manipulation:
    - P(S=1 | Y=1)>P(S=1)
    - P(S=1 | do(Y=1))=P(S=1)

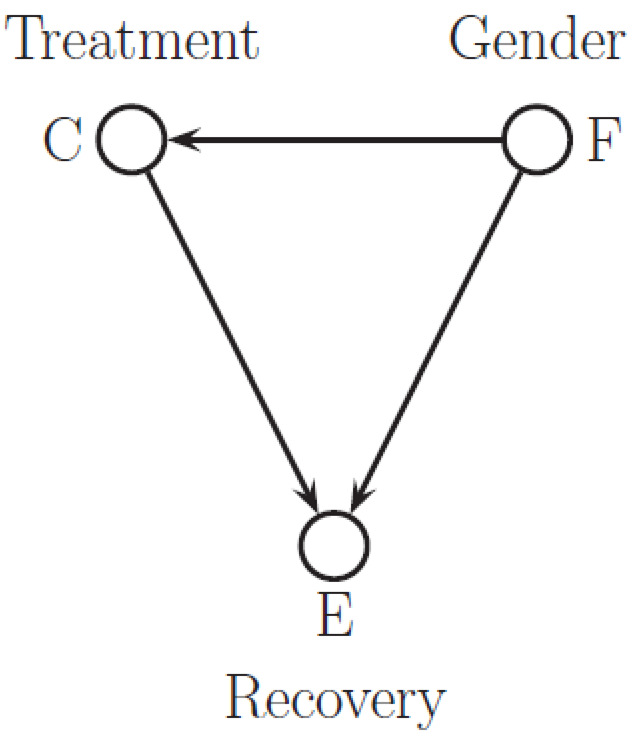
GRAPH SURGERY



SIMPSON'S PARADOX

- Any statistical relationship between two variables can be reversed by including additional factors the analysis
- $P(E|C) > P(E|\neg C)$
- Include additional factor F:
- $P(E|C, F) < P(E|\neg C, F)$
- $P(E|C, \neg F) < P(E|\neg C, \neg F)$
- The statement is not interpreted properly!
- Should be $P(E|do(C)) > P(E|do(\neg C))$

SIMPSON'S PARADOX



Suppose

$$p(E|do(C), F) < p(E|do(\neg C), F)$$

$$p(E|do(C), \neg F) < p(E|do(\neg C), \neg F)$$

We show

$$p(E|do(C)) < p(E|do(\neg C))$$

SIMPSON'S PARADOX

- Proof:
  - $p(F|do(C)) = p(F|do(\neg C)) = p(F)$
  - $p(E|do(C)) = p(E|do(C), F)p(F|do(C)) + p(E|do(C), \neg F)p(\neg F|do(C))$
    $= p(E|do(C), F)p(F) + p(E|do(C), \neg F)p(\neg F)$
  - $p(E|do(\neg C)) = p(E|do(\neg C), F)p(F) + p(E|do(\neg C), \neg F)p(\neg F)$
  - Given $p(E \mid do(C), F) < p(E \mid do(\neg C), F)$ and $p(E \mid do(C), \neg F) < p(E \mid do(\neg C), \neg F)$
  - Conclusion: $p(E|do(C)) < p(E|do(\neg C))$

LEARNING CAUSAL DAG STRUCTURES

- Learning from observational data
  - Learn an PDAG from data
  - Enumerate all the DAGs from the PDAG equivalence class
  - Apply Pearl's do-calculus to compute the magnitude of each causal effect pair
  - Take the minimum of these effects as the lower bound

- Learning from interventional data
  - Control some variables and measure the consequences
  - First skipping over intervention cases
  - Then adding the intervention nodes with constraints

CONTENTS

- Structure learning for knowledge discovery
- Learning tree structures
- Learning DAG structures
- Learning DAG structure with latent variables
- Learning causal DAGs
- **Learning undirected Gaussian graphical models**
- Learning undirected discrete graphical models

LEARNING UNDIRECTED GAUSSIAN GRAPHICAL MODELS

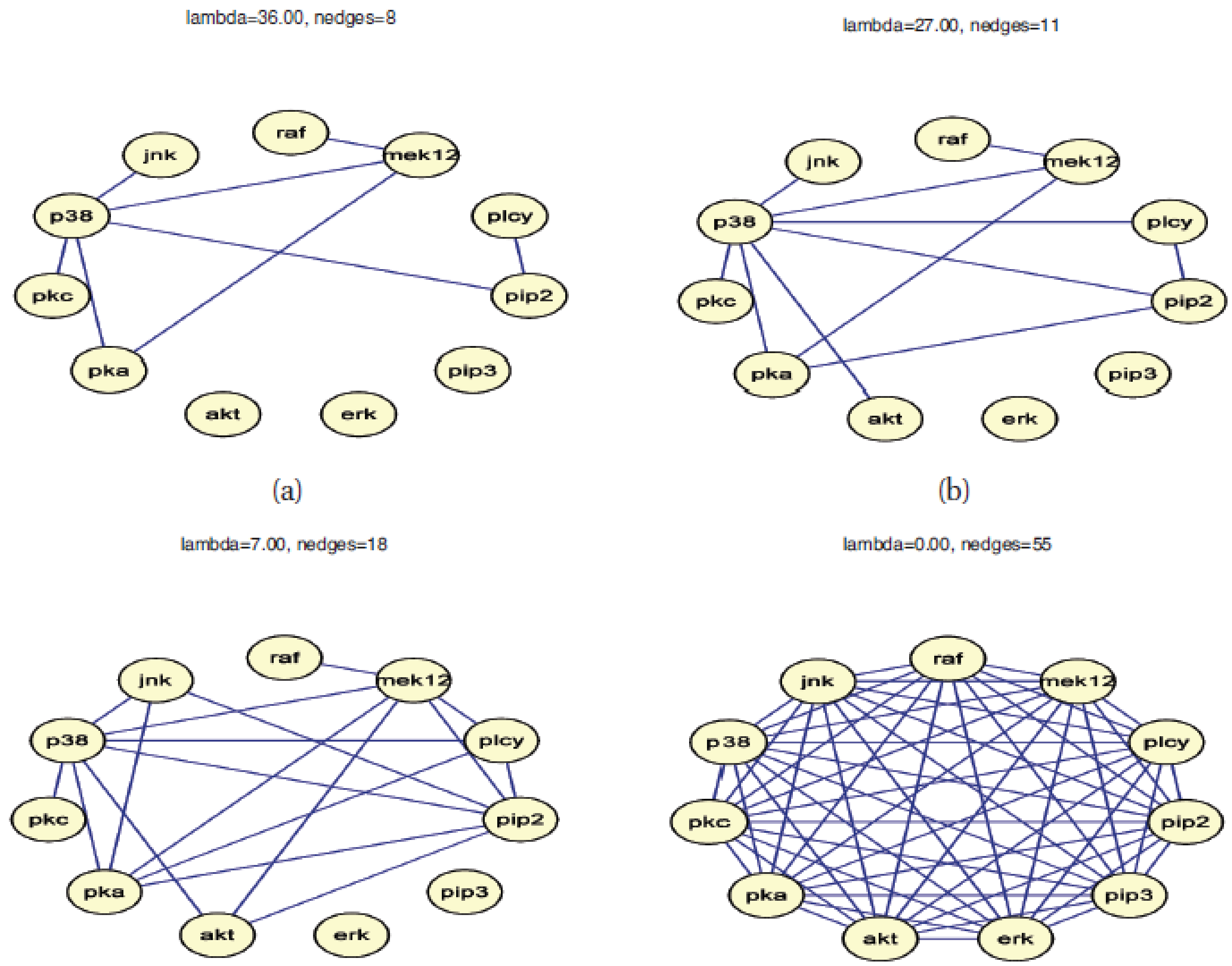
- *MLE for a GGM - covariance estimation*
- $l(\Omega) = \log|\Omega| - tr(S\Omega)$
- $\nabla l(\Omega) = \Omega^{-1} - S$
- Constraints:
  - $\Omega_{st} = 0 \; if \; G_{st} = 0$
  - Ω is positive definite
- Property: $\Sigma_{st} = S_{st}$ if $\mathrm{G_{st}} = 1$ or $\mathrm{s} = \mathrm{t}$

GRAPHICAL LASSO

- $J(\Omega) = -\log|\Omega| + tr(S\Omega) + \lambda||\Omega||_1$
- Convex but non-smooth
- Coordinate descent algorithm

- Shooting algorithm for lasso $\quad \arg\min_x \|Ax - y\|_2^2 + \lambda\|x\|_1$

$$\begin{aligned} S_j &= A_{(:,j)}^T(Ax) - 2(y^TA)_j + (A^TA)_{j,j}x_j^{t-1} \\ x_j^t &\leftarrow \operatorname{sign}(S_j)(|S_j| - \lambda)_+ \end{aligned}$$


lambda=36.00, nedges=8
lambda=27.00, nedges=11
lambda=7.00, nedges=18
lambda=0.00, nedges=55
raf
mek12
jnk
p38
plcy
pkc
pip2
pka
pip3
akt
erk

(a)

(b)

CONTENTS

- Structure learning for knowledge discovery
- Learning tree structures
- Learning DAG structures
- Learning DAG structure with latent variables
- Learning causal DAGs
- Learning undirected Gaussian graphical models
- **Learning undirected discrete graphical models**

LEARNING UNDIRECTED DISCRETE GRAPHICAL MODELS

- Grapical lasso for CRF

$$\psi_t(y_t, \mathbf{x}) = \begin{pmatrix} \mathbf{v}_{t1}^T\mathbf{x} \\ \mathbf{v}_{t2}^T\mathbf{x} \\ \mathbf{v}_{t3}^T\mathbf{x} \end{pmatrix}, \quad \psi_{st}(y_s, y_t, \mathbf{x}) = \begin{pmatrix} \mathbf{w}_{t11}^T\mathbf{x} & \mathbf{w}_{st12}^T\mathbf{x} & \mathbf{w}_{st13}^T\mathbf{x} \\ \mathbf{w}_{st21}^T\mathbf{x} & \mathbf{w}_{st22}^T\mathbf{x} & \mathbf{w}_{st23}^T\mathbf{x} \\ \mathbf{w}_{st31}^T\mathbf{x} & \mathbf{w}_{st32}^T\mathbf{x} & \mathbf{w}_{st33}^T\mathbf{x} \end{pmatrix}$$

$$\begin{aligned} J &= -\sum_{i=1}^{N}\left[\sum_{t}\log\psi_t(y_{it}, \mathbf{x}_i, \mathbf{v}_t) + \sum_{s=1}^{V}\sum_{t=s+1}^{V}\log\psi_{st}(y_{is}, y_{it}, \mathbf{x}_i, \mathbf{w}_{st})\right] \\ &\quad +\lambda_1\sum_{s=1}^{V}\sum_{t=s+1}^{V}||\mathbf{w}_{st}||_p + \lambda_2\sum_{t=1}^{V}||\mathbf{v}_t||_2^2 \end{aligned}$$

- Thin junction tree – bound the treewidth

SUMMARY

- Structure learning for knowledge discovery
- Learning tree structures
- Learning DAG structures
- Learning DAG structure with latent variables
- Learning causal DAGs
- Learning undirected Gaussian graphical models
- Learning undirected discrete graphical models

- Thanks!