グラフィカルモデルの推定 – パラメータ推定と構造学習

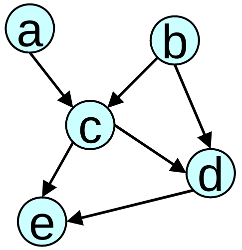
Kenji Fukumizu
The Institute of Statistical Mathematics

計算推論科学概論 II （2010年度, 後期）

Working with Graphical Models

- Determining structure
  - Structure given by modeling

    e.g. Mixture model, HMM
  - Structure learning



structure → Part 4

- Parameter estimation
  - Parameter given by some knowledge
  - Parameter estimation with data such as MLE or Bayesian estimation

    → Part 4

$p(X_c \mid X_a)$

$X_c \setminus X_a$	1	2	3
1	0.2	0.3	0.4
2	0.8	0.7	0.6

parameter

- Inference
  - Computation of posterior and marginal probabilities

    (Already seen in Part 3.)

Parameter Estimation

Statistical Estimation

- Estimation from data

  Statistical model with a parameter: $p(X \mid \theta)$ θ: parameter

  I.i.d. Data: $D = (X_1, X_2, \ldots, X_N)$

  - Maximum likelihood estimation

$$\hat{\theta} = \arg\max_{\theta} L(\theta), \qquad L(\theta) = \prod_{i=1}^{N} p(X_i \mid \theta)$$

Likelihood function

or

$$\hat{\theta} = \arg\max_{\theta} \ell(\theta)$$

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{N} \log p(X_i \mid \theta)$$

Log likelihood function

Statistical Estimation

- Bayesian estimation
  - Distribution of the parameter θ is estimated

    Prior probability $p(\theta)$ → posterior probability $p(\theta \mid D)$

    Bayes' rule (Bayes' thoerem)

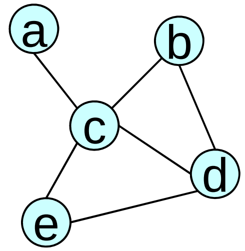
$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{p(D)} = \frac{\prod_{i=1}^{N} p(X_i \mid \theta) p(\theta)}{\int \prod_{i=1}^{N} p(X_i \mid \theta) p(\theta) d\theta}$$

  - Maximum a posteriori (MAP) estimation

$$\hat{\theta}_{MAP} = \arg\max_{\theta}\, p(\theta \mid D)$$

Contingency Table（分割表）

- ML estimation for discrete variables



$$X_a \in \{1, \ldots, M\} \qquad X_b \in \{1, \ldots, L\}$$

$$D = (X_a^{(1)}, X_b^{(1)}), \ldots, (X_a^{(N)}, X_b^{(N)})$$ i.i.d. sample

$X_b \backslash X_a$	1	2	3
1	12	18	4
2	6	9	14

N_{ij}: Number of counts

$p(X_a, X_b)$

$X_b \backslash X_a$	1	2	3
1	p_{11}	p_{12}	p_{13}
2	p_{21}	p_{22}	p_{22}

Estimation of probabilities

ML estimator $$\hat{p}_{ij} = \frac{N_{ij}}{N}$$

Bayesian Estimation: Discrete Case

- Bayesian estimation for discrete variables

Model: $p(X_a, X_b \mid \theta)$

$$p(X_a = i, X_b = j \mid \theta) = \theta_{ij}, \qquad \theta = \left(\theta_{ij}\right) \in \Delta_{ML-1}$$

$$\Delta_{K-1} \equiv \{\theta \in \mathbf{R}^K \mid \theta_i \geq 0\ (\forall i), \textstyle\sum_{i=1}^K \theta_i = 1\}$$

Prior: $\pi(\theta)$ on Δ_{ML-1}

Likelihood: $p(D \mid \theta) = \prod_{n=1}^{N} p(X_a^{(n)}, X_b^{(n)} \mid \theta) = \prod_{i,j} \theta_{ij}^{N_{ij}}$ Multinomial

Bayesian estimation:

$$p(\theta \mid D) = \frac{p(D,\theta)}{p(D)} = \frac{p(D \mid \theta)\pi(\theta)}{\int_\Delta p(D \mid \theta)\pi(\theta)d\theta} = \frac{\prod_{i,j} \theta_{ij}^{N_{ij}} \pi(\theta)}{\underline{\int_\Delta \theta_{ij}^{N_{ij}} \pi(\theta)d\theta}}$$

This integral is difficult to compute in general.

Dirichlet Distribution

- Dirichlet distribution
  - Density function of K-dimensional Dirichlet distribution

$$\mathrm{Dir}(\theta \mid \alpha_1,\dots,\alpha_K) = \frac{\Gamma(\sum_{j=1}^{K}\alpha_j)}{\prod_{j=1}^{K}\Gamma(\alpha_j)}\prod_{j=1}^{K}\theta_j^{\alpha_j-1} \quad \propto \prod_{j=1}^{K}\theta_j^{\alpha_j-1}$$

on $\quad \Delta_{K-1} = \{\theta \in \mathrm{R}^K \mid \theta_j \geq 0, \sum_{j=1}^{K}\theta_j = 1\}$

where

$(\alpha_1,\dots,\alpha_K)$: parameter $(\alpha_j > 0)$

$\Gamma(\alpha)$: Gamma function $\quad \Gamma(\alpha) = \int_0^\infty t^{\alpha-1}e^{-t}dt$

$\Gamma(\alpha) = (\alpha-1)\Gamma(\alpha-1)$ for $\alpha > 1$

$\Gamma(n) = (n-1)!$ for a positive integer n.

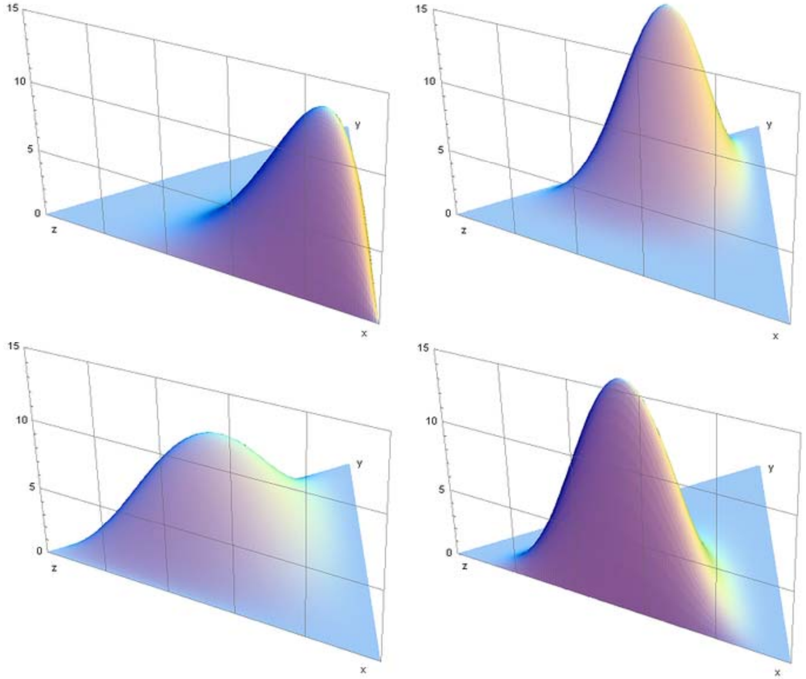
Dirichlet Distribution

α = (6,2,2)

α = (3,7,5)

α = (2,3,4)

α = (6,2,6)



- Expectation

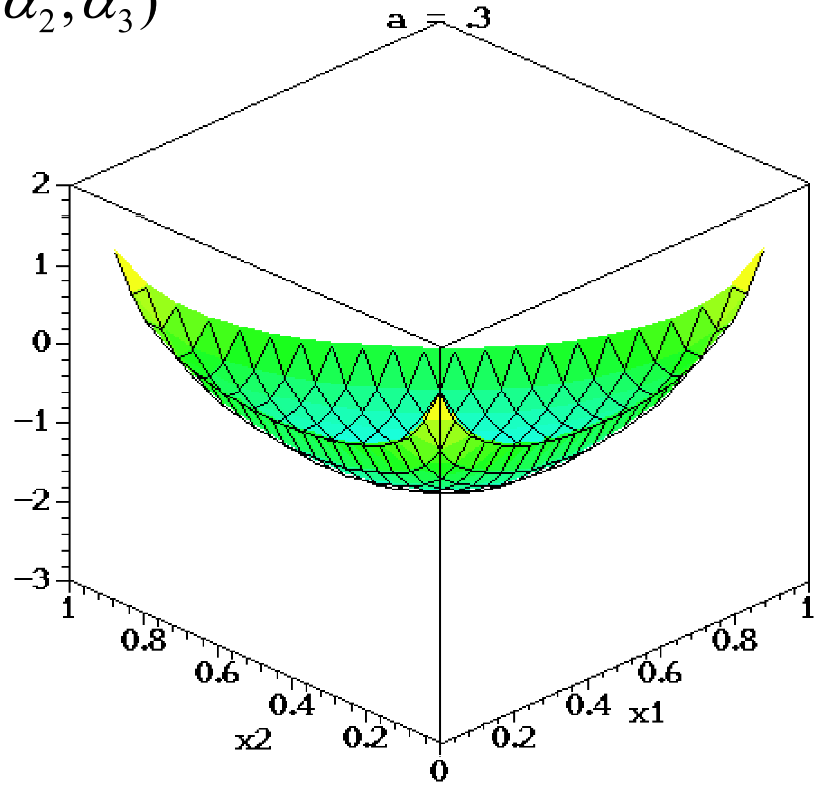
$$E[\theta_i] = \frac{\alpha_i}{\sum_{j=1}^{K} \alpha_j}$$

[Exercise]

  - The mean point is proportional to the vector α.
  - The mean point is a stable point (i.e. differential = 0), and it may be either maximum or minimum.

Dirichlet Distribution

$\text{Dir}(\theta \mid \alpha_1, \alpha_2, \alpha_3)$



K=3. *α* = *b*(1, 1, 1) from *b*=0.3 to 2.0.

Bayesian Inference with Dirichlet Prior

- Dirichlet distribution works as a prior to multinomial distribution.
  Posterior is also Dirichlet -- conjugate prior

  Data: $D = (X^{(1)}, \ldots, X^{(N)}) \qquad N_k := \left|\{i \mid X^{(i)} = k\}\right| \quad (k = 1, \ldots, K)$

  Posterior:

$$p(\theta \mid D) = \frac{p(D \mid \theta)\mathrm{Dir}(\theta \mid \alpha)}{\int_\Delta p(D \mid \theta)\mathrm{Dir}(\theta \mid \alpha) d\theta} = \frac{\prod_k \theta_k^{N_k} \mathrm{Dir}(\theta \mid \alpha)}{\int_\Delta \theta_k^{N_k} \mathrm{Dir}(\theta \mid \alpha) d\theta} = \underline{\mathrm{Dir}(\theta \mid \tilde{\alpha})}$$

$$\tilde{\alpha} = (N_1 + \alpha_1, \ldots, N_K + \alpha_K)$$

α works as a prior count.

- MAP estimator

$$\hat{\theta}_{MAP} = \frac{\tilde{\alpha}_i}{\sum_{j=1}^{K} \tilde{\alpha}_j} = \frac{N_i + \alpha_i}{N + \alpha_1 + \cdots \alpha_K}$$

Bayesian Inference with Dirichlet Prior

Proof.

$$p(\theta \mid D) \propto \prod_{j=1}^{K} \theta_j^{N_j} \mathrm{Dir}(\theta \mid \alpha) \propto \prod_{j=1}^{K} \theta_j^{N_j + \alpha_j - 1}$$

By the normalization, the right hand side must be $\mathrm{Dir}(\theta \mid \tilde{\alpha})$.

EM Algorithm
for Models with Hidden Variables

ML Estimation with Hidden Variable

- Statistical model with hidden variables
  - Suppose we can assume hidden (unobservable) variables in addition to observable variables.

$$p(X, Z \mid \theta)$$

X: observable variable
Z: hidden variable
θ: parameter

  - We have data only for observable variables: $D = (X_1, X_2, \ldots, X_N)$
    The ML estimation must be done with X

$$\sum_{n=1}^{N} \log p(X_n \mid \theta) = \sum_{n=1}^{N} \log \left(\sum_{Z_n} p(X_n, Z_n \mid \theta) \right)$$

But, this maximization is often difficult.

  - Probability of (X, Z) is sometimes easier to handle than that of X.

ML Estimation with Hidden Variable

- Example: Gaussian mixture model

  With hidden variable: $p(X, Z \mid \theta) = p(Z \mid \pi)\phi(x \mid \mu_j, \Sigma_j)$

  $$Z \longrightarrow X$$

  Z takes values in {1,...,K}: component

  $\theta = (\pi, \mu_1, \Sigma_1, \ldots, \mu_K, \Sigma_K)$

  Marginal of X: $$p(x \mid \theta) = \sum_{j=1}^{K} \pi_j \phi(x \mid \mu_j, \Sigma_j)$$

- ML estimation

$$\max_{\theta} \sum_{n=1}^{N} \log p(X_n \mid \theta) = \max_{\theta} \sum_{n=1}^{N} \log\left(\sum_{j=1}^{K} \pi_j \phi(X_n \mid \mu_j, \Sigma_j) \right)$$

π_j and (μ_j, Σ_j) are coupled → difficult to solve analytically.

Estimation with Complete Data

- Complete data
  - Suppose $Z_1, \ldots, Z_N$ were known.

$$D_c = \{(X_1, Z_1), \ldots, (X_N, Z_N)\}$$: complete data

ML estimation with D_c is often easier than estimation with D.

$$\max \ell_c(D_c \mid \theta),$$

where

$$\ell_c(D_c \mid \theta) = \sum_{n=1}^{N} \log p(X_n, Z_n \mid \theta)$$ Complete log likelihood

Estimation with Complete Data

- Example: Mixture of Gaussian

  Redefine the hidden variable Z by K dimensional binary vector:

$$\boxed{p(X, Z \mid \theta) = \prod_{a=1}^{K} \{\pi_a \phi(x \mid \mu_a, \Sigma_a)\}^{Z_a}}$$

$Z = (Z_1,\dots,Z_K)$ takes values in

$$\{ (1,0,0,\dots,0),\ (0,1,0,\dots,0),\ \cdots\ (0,0,0,\dots,1) \}$$ K class

Note: $p(X \mid \theta) = \sum_{Z} p(X, Z \mid \theta) = \sum_{a=1}^{K} \pi_a \phi(x \mid \mu_a, \Sigma_a)$

Estimation with Complete Data

ML estimation with complete data:

$$\sum_{n=1}^{N} \log p(X_n, Z_n \mid \theta) = \sum_{n=1}^{N} \log\left(\prod_{i=1}^{K} \{\pi_i \phi(X_n \mid \mu_i, \Sigma_i)\}^{Z_i^n} \right)$$

$$= \sum_{n=1}^{N} \sum_{i=1}^{K} Z_i^n \{\log \pi_i + \log \phi(X_n \mid \mu_i, \Sigma_i)\}$$

π_j and (μ_j, Σ_j) are decoupled → they can be maximized separately.

$$\begin{cases} \max\limits_{\pi} \sum_{n=1}^{N} \sum_{i=1}^{K} Z_i^n \log \pi_i \quad \text{subj. to} \quad \sum_{i=1}^{K} \pi_i = 1 \\ \max\limits_{\mu,\Sigma} \sum_{n=1}^{N} \sum_{i=1}^{K} Z_i^n \log \phi(X_n \mid \mu_i, \Sigma_i) \end{cases}$$

Maximization is easy.

But, the complete data is not available in practice!

Expected Complete Log Likelihood

- Use expected complete log likelihood instead of complete log likelihood.

- Complete log likelihood

$$\ell_c(D_c \mid \theta) = \sum_{n=1}^{N} \log p(X_n, Z_n \mid \theta)$$

- Expected complete log likelihood
  - Suppose we have a current guess $\hat{\theta}^{(t)}$

    Use expectation w.r.t. $p(Z_n \mid X_n, \hat{\theta}^{(t)})$

$$\langle \ell_c(D_c \mid \theta) \rangle_{\hat{\theta}^{(t)}} = \sum_{n=1}^{N} \sum_{Z_n} p(Z_n \mid X_n, \hat{\theta}^{(t)}) \log p(X_n, Z_n \mid \theta)$$

Maximize θ of $\langle \ell_c(D_c \mid \theta) \rangle_{\hat{\theta}^{(t)}}$

EM Algorithm

Initialization

Initialize $\theta = \theta^{(0)}$ by some method.

$t = 0$.

Repeat the following steps until stopping criterion is satisfied.

E-step

Compute the expected complete log likelihood $\langle \ell_c(D_c \mid \theta) \rangle_{\hat{\theta}^{(t)}}$

M-step

Maximize θ of $\langle \ell_c(D_c \mid \theta) \rangle_{\hat{\theta}^{(t)}}$

$$\hat{\theta}^{(t+1)} = \arg\max_{\theta} \langle \ell_c(D_c \mid \theta) \rangle_{\hat{\theta}^{(t)}}$$

- Computational difficulty of M-step depends on the model.

EM Algorithm for Gaussian Mixture

- Complete log likelihood

$$\ell_c(D_c \mid \theta) = \sum_{n=1}^{N}\sum_{i=1}^{K} Z_i^n \{\log \pi_i + \log \phi(X_n \mid \mu_i, \Sigma_i)\}$$

- Expected complete log likelihood

$$\tau_i^{n(t)} = E[Z_i^n \mid X_n, \hat{\theta}^{(t)}] = p(Z_i^n = 1 \mid X_n, \hat{\theta}^{(t)}) = \frac{p(X_n, Z_i^n = 1 \mid \hat{\theta}^{(t)})}{p(X_n \mid \hat{\theta}^{(t)})}$$

$$= \frac{\hat{\pi}_i^{(t)} \phi(X_n \mid \hat{\mu}_i^{(t)}, \hat{\Sigma}_i^{(t)})}{\sum_{j=1}^{K} \hat{\pi}_j^{(t)} \phi(X_n \mid \hat{\mu}_j^{(t)}, \hat{\Sigma}_j^{(t)})}$$

Ratio of contribution of X_n to the i-th component.

- E-step

$$\langle \ell(D_c \mid \theta) \rangle_{\hat{\theta}^{(t)}} = \sum_{n=1}^{N}\sum_{i=1}^{K} \underline{\tau_i^{n(t)}} \{\log \pi_i + \log \phi(X_n \mid \mu_i, \Sigma_i)\}$$

EM Algorithm for Gaussian Mixture

- M-step

$$\hat{\pi}_i^{(t+1)} = \frac{1}{N}\sum_{n=1}^{N}\tau_i^{n(t)}$$

$$\hat{\mu}_i^{(t+1)} = \frac{\sum_{n=1}^{N}\tau_i^{n(t)}X_n}{\sum_{n=1}^{N}\tau_i^{n(t)}}$$ weighted mean

$$\hat{\Sigma}_i^{(t+1)} = \frac{\sum_{n=1}^{N}\tau_i^{n(t)}(X_n-\hat{\mu}_i^{(t)})(X_n-\hat{\mu}_i^{(t)})^T}{\sum_{n=1}^{N}\tau_i^{n(t)}}$$ weighted covariance matrix

(Proof omitted. Exercise)

EM Algorithm for Gaussian Mixture

- Meaning of τ

$Z_n^{\,i}$ (if observed)

n \ i	1	2	3	K
1	0	1	0	0
2	0	0	0	1
3	1	0	0	0
	$\vdots$			$\vdots$
N	0	0	0	1

$$\tau_n^{i(t)} = E\left[Z_n^i \mid X_n, \hat{\theta}^{(t)} \right]$$

n \ i	1	2	3	K	SUM
1	0.1	0.7	0	0.2	➡ 1
2	0.2	0.1	0.2	0.5	➡ 1
3	0.8	0.1	0.05	0.05	➡ 1
	$\vdots$			$\vdots$	$\vdots$
N	0.13	0.11	0.06	0.7	➡ 1

Properties of EM Algorithm

- EM converges quickly for many problems.
- Monotonic increase of likelihood of X is guaranteed (discussed later).
- EM may be trapped by local optima.
- The solution depends strongly on the initial state.
- EM algorithm can be applied to any model with hidden variables. Missing value, etc.

Demonstration

- Web site for Gaussian mixture demo:

  http://staff.aist.go.jp/s.akaho/MixtureEMj.html

Theoretical Justification of EM

Theoretical Justification of EM

- EM as likelihood maximization

The goal is to maximize the (incomplete) log likelihood, not the expected complete log likelihood.

$q(Z \mid X)$: arbitrary p.d.f. of Z, may depend on X.

Define an auxiliary function $L(q,\theta)$ by

$$L(q,\theta) = \sum_Z q(Z \mid X) \log \frac{p(X, Z \mid \theta)}{q(Z \mid X)}.$$

<u>Theorem 1</u>

E-step: $q^{(t+1)} = \arg\max_q L(q, \hat{\theta}^{(t)})$ (and compute $\langle \ell_c(D_c \mid \theta) \rangle_{q^{(t+1)}}$)

M-step: $\hat{\theta}^{(t+1)} = \arg\max_\theta L(q^{(t+1)}, \theta)$

Alternating optimization w.r.t. q and θ.

Theoretical Justification of EM

<u>Proposition 1</u> (L and likelihood of X)

For any $q(Z \mid X)$ and θ, the log likelihood of X is decomposed as

$$\ell(X \mid \theta) = L(q,\theta) + KL(q(Z \mid X) \,\|\, p(Z \mid X,\theta))$$

In particular,

$$\ell(X \mid \theta) \geq L(q,\theta) \qquad \text{for all } q \text{ and } \theta,$$

and the equality holds if and only if $q = p(Z \mid X,\theta)$.

Proof) $\ell(\theta \mid X) - L(q,\theta)$

$$= \underbrace{\sum_Z q(Z \mid X)}_{=1} \log p(X \mid \theta) - \sum_Z q(Z \mid X) \log \frac{p(X,Z \mid \theta)}{q(Z \mid X)}$$

$$= \sum_Z q(Z \mid X) \log \frac{p(X \mid \theta) q(Z \mid X)}{p(X,Z \mid \theta)}$$

$$= \sum_Z q(Z \mid X) \log \frac{q(Z \mid X)}{p(Z \mid X,\theta)}$$

Theoretical Justification of EM

Proposition 2 (L and expected complete likelihood)

$$L(q,\theta) = \langle \ell_c(X,Z \mid \theta) \rangle_q - \sum_Z q(Z \mid X) \log q(Z \mid X)$$

proof)

$$\begin{aligned}
\langle \ell(X,Z \mid \theta) \rangle_q &= \sum_Z q(Z \mid X) \log p(X,Z \mid \theta) \\
&= \sum_Z q(Z \mid X) \log \frac{p(X,Z \mid \theta) q(Z \mid X)}{q(Z \mid X)} \\
&= \sum_Z q(Z \mid X) \log \frac{p(X,Z \mid \theta)}{q(Z \mid X)} + \sum_Z q(Z \mid X) \log q(Z \mid X) \\
&= L(q,\theta) + \sum_Z q(Z \mid X) \log q(Z \mid X)
\end{aligned}$$

Theoretical Justification of EM

- Proof of Theorem 1
  - E-step:

    From Proposition 1,

$$\underbrace{\ell(X \mid \hat{\theta}^{(t)})}_{\text{independent of } q} = \underbrace{L(q, \hat{\theta}^{(t)})}_{\text{maximize}} + \underbrace{KL(q(Z \mid X) \,\|\, p(Z \mid X, \hat{\theta}^{(t)}))}_{\text{minimize}}$$

independent of q maximize $\Leftrightarrow$ minimize

$$\Rightarrow \quad p(Z \mid X, \hat{\theta}^{(t)}) = \arg\max_{q} L(q, \hat{\theta}^{(t)})$$

  - M-step:

    From Proposition 2,

$$L(q^{(t+1)}, \theta) = \left\langle \ell_c(X, Z \mid \theta) \right\rangle_{p(Z \mid X, \hat{\theta}^{(t)})} - (\text{const. w.r.t. } \theta)$$

M-step is

$$\max_{\theta} L(q^{(t+1)}, \theta)$$

Theoretical Justification of EM

- Monotonic increase of likelihood by EM

<u>Theorem</u>

$$\ell(X \mid \hat{\theta}^{(t)}) \leq \ell(X \mid \hat{\theta}^{(t+1)}) \qquad \text{for all } t.$$

Proof)

$$\ell(X \mid \hat{\theta}^{(t)}) = L(q^{(t+1)}, \hat{\theta}^{(t)}) \qquad \text{(E-step, Prop.1)}$$

$$\leq L(q^{(t+1)}, \hat{\theta}^{(t+1)}) \qquad \text{(M-step)}$$

$$\leq \ell(X \mid \hat{\theta}^{(t+1)}) \qquad \text{(Prop.1)}$$

Remarks on EM Algorithm

- EM always increases the likelihood of observable variables, but there are no theoretical guarantees of global maximization.
  In general, it can converge only to a local maximum.
- There is a sufficient condition of convergence by Wu (1983).
- Practically, EM converges very quickly.

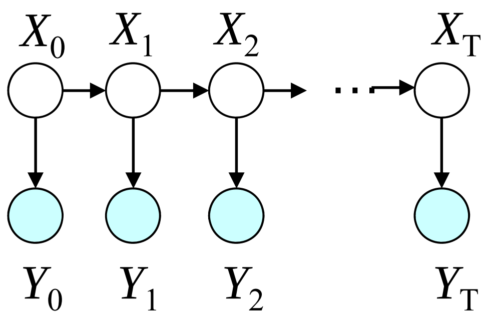
- For Gaussian mixture model,
  - If the mean and variance are its parameters, the likelihood function can take an arbitrary large value. There is no global maximum of likelihood.
  - EM often finds a reasonable local optimum by a good choice of initialization.
  - The results depend much on the initialization.

- Further readings:
  - *The EM Algorithm and Extensions* (McLachlan & Krishnan 1997)
  - *Finite Mixture Models* (McLachlan & Peel 2000)

EM Algorithm for Hidden Markov Model

Maximum Likelihood for HMM

- Parametric model of Gaussian HMM



$$p(X,Y) = p(X_0)\prod_{t=0}^{T-1} p(X_{t+1} \mid X_t)\prod_{t=0}^{T} p(Y_t \mid X_t)$$

$$p(X_0 = j) = \pi_j$$ initial probability

$$p(X_{t+1} = j \mid X_t = i) = A_{ij}$$ transition matrix (time invariant)

$$p(Y_t \mid X_t = j) = \phi(y_t; \mu_j, \Sigma_j)$$ Gaussian with mean μ_j and covariance Σ_j

parameter: $\theta = (\pi, (A_{ij}), \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K)$

$$p(Y \mid \theta) = \sum_{X_0} \cdots \sum_{X_T} \pi_{X_0} \prod_{t=0}^{T-1} A_{X_{t-1}X_t} \prod_{t=0}^{T} \phi(y_t \mid \mu_{X_t}, \Sigma_{X_t})$$

max log $p(Y \mid \theta)$ is difficult.

EM for HMM

- Complete likelihood

$$\ell_c(Y,X\mid\theta)=\log p(Y,X\mid\theta)$$

$$=\log\left(\pi_{X_0}\prod_{t=0}^{T-1}A_{X_tX_{t+1}}\prod_{t=0}^{T}\phi(Y_t\mid\mu_{X_t},\Sigma_{X_t})\right)$$

log ($\Pi_t\ \alpha_t$) is easy.

$$=\log\pi_{X_0}+\sum_{t=0}^{T-1}A_{X_tX_{t+1}}$$

$$+\sum_{t=0}^{T}\left\{-\frac{1}{2}(Y_t-\mu_{X_t})^T\Sigma_{X_t}^{-1}(Y_t-\mu_{X_t})-\frac{1}{2}\log\det\Sigma_{X_t}-\frac{m}{2}\log(2\pi)\right\}$$

$$=\sum_{j=1}^{K}\delta_{jX_0}\log\pi_j+\sum_{i,j=1}^{K}\sum_{t=0}^{T-1}\delta_{jX_{t+1}}\delta_{iX_t}A_{ij}$$

$$+\sum_{j=1}^{K}\sum_{t=0}^{T}\delta_{jX_t}\left\{-\frac{1}{2}(Y_t-\mu_j)^T\Sigma_j^{-1}(Y_t-\mu_j)-\frac{1}{2}\log\det\Sigma_j-\frac{m}{2}\log(2\pi)\right\}$$

EM for HMM

- Expected complete likelihood

Suppose we already have an estimate $\hat{\theta}^{(n)}$ (n: index for iteration)

$$\left\langle \ell_c(Y,X\mid\theta)\right\rangle_{\hat{\theta}^{(n)}} = \sum_X p(X\mid Y,\hat{\theta}^{(n)})\log p(Y,X\mid\theta)$$

It requires

$$\left\langle \delta_{jX_t}\right\rangle_{\hat{\theta}^{(n)}} = \sum_{X_t=1}^{K} p(X_t\mid Y,\hat{\theta}^{(n)})\delta_{jX_t} = p(X_t=j\mid Y,\hat{\theta}^{(n)}) \equiv \gamma_t^{j(n)}$$

$$\left\langle \delta_{iX_t}\delta_{jX_{t+1}}\right\rangle_{\hat{\theta}^{(n)}} = \sum_{X_t=1}^{K}\sum_{X_{t+1}=1}^{K} p(X_t,X_{t+1}\mid Y,\hat{\theta}^{(n)})\delta_{iX_t}\delta_{jX_{t+1}}$$

$$= p(X_t=i,X_{t+1}=j\mid Y,\hat{\theta}^{(n)}) \equiv \xi_{t,t+1}^{i,j(n)}$$

$\gamma_t^{j(n)} = p(X_t=j\mid Y,\hat{\theta}^{(n)})$ and $\xi_{t,t+1}^{i,j(n)} = p(X_t=i,X_{t+1}=j\mid Y,\hat{\theta}^{(n)})$

can be computed by the forward-backward algorithm.

EM for HMM – Baum-Welch Algorithm

- E-step
  - Forward-backward to compute $\gamma_t^{j(n)}$ and $\xi_{t,t+1}^{i,j(n)}$.
  - Expected complete log likelihood

$$\left\langle \ell_c(Y, X \mid \theta) \right\rangle_{\hat{\theta}^{(n)}} = \sum_{j=1}^{K} \gamma_0^{j(n)} \log \pi_j + \sum_{i,j=1}^{K} \sum_{t=0}^{T-1} \xi_{t,t+1}^{i,j(n)} A_{ij}$$

$$+ \sum_{j=1}^{K} \sum_{t=0}^{T} \gamma_t^{j(n)} \left\{ -\frac{1}{2} (Y_t - \mu_j)^T \Sigma_j^{-1} (Y_t - \mu_j) - \frac{1}{2} \log \det \Sigma_j - \frac{m}{2} \log(2\pi) \right\}$$

- M-step

$$\hat{\pi}_j^{(n+1)} = \gamma_0^{j(n)}, \qquad \hat{A}_{i,j}^{(n+1)} = \frac{\sum_{t=0}^{T-1} \xi_{t,t+1}^{i,j(n)}}{\sum_{k=1}^{K} \sum_{t=0}^{T-1} \xi_{t,t+1}^{i,k(n)}} = \frac{\sum_{t=0}^{T-1} \xi_{t,t+1}^{i,j(n)}}{\sum_{t=0}^{T-1} \gamma_t^{i(n)}}$$

$$\hat{\mu}_i^{(n+1)} = \frac{\sum_{t=0}^{T-1} \gamma_t^{i(n)} Y_t}{\sum_{t=0}^{T-1} \gamma_t^{i(n)}}, \qquad \hat{\Sigma}_i^{(n+1)} = \frac{\sum_{t=0}^{T-1} \gamma_t^{i(n)} (Y_t - \mu_i^{(n+1)})(Y_t - \mu_i^{(n+1)})^T}{\sum_{t=0}^{T-1} \gamma_t^{i(n)}}$$

c.f. EM for Gaussian mixture

Summary: Parameter learning

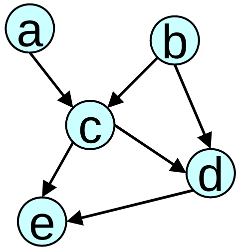
- Discrete variables without hidden variables
  - Maximum likelihood estimation is easy by frequencies.
  - Bayesian estimation is often done with Dirichlet prior.

- Discrete variables with hidden variables
  - Maximum likelihood estimation can be done with EM algorithm.
  - Bayesian approach → computational difficulty.
    Some technique is needed, *e.g. variational method*.

Structure Learning

Working with Graphical Models

- Determining structure
  - Structure given by modeling
    e.g. Mixture model, HMM
  - Structure learning



structure → Part 4

- Parameter estimation
  - Parameter given by some knowledge
  - Parameter estimation with data
    such as MLE or Bayesian estimation
    → Part 4

$p(X_c \mid X_a)$

$X_c \setminus X_a$	1	2	3
1	0.2	0.3	0.4
2	0.8	0.7	0.6

parameter

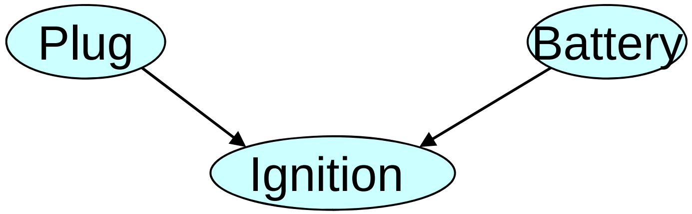
- Inference
  - Computation of posterior and marginal probabilities
    (Already seen in Part 3.)

How to determine a network?

- Prior knowledge

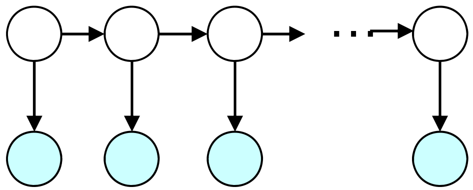
A graphical model may be given by the prior knowledge on the problem.

e.g.1) Diagnosis system



The problem is to estimate the probabilities (parameters).

e.g.2) HMM



- Structure learning

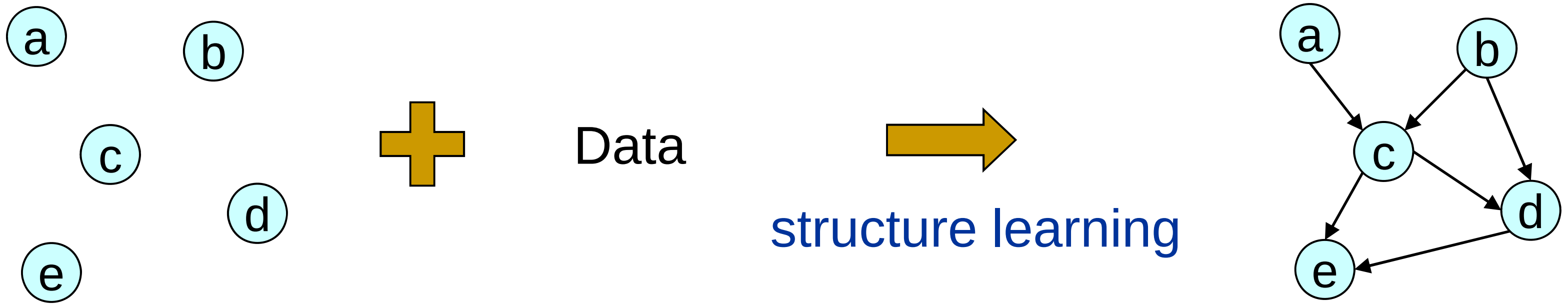
If it is difficult to assume an appropriate model, the graph structure must be *learned* from data.

Structure Learning

Variables: $X_1, \ldots, X_m$

Data: $(X_1^{(1)}, \ldots, X_m^{(1)}), \ldots, (X_1^{(N)}, \ldots, X_m^{(N)})$

Output of structure learning = a directed / undirected graph associated with the probability of $(X_1, \ldots, X_m)$.



Difficulty: the number of possible directed graphs = $3^{m(m-1)/2}$

The search space is very large.

Learning of Directed Graph

- Score-based method
  - Use a global score to match a graph and data.
  - Problem: Optimization in huge search space.
  - Able to use informative prior on graphs.
  - Usually, discrete variables are assumed.
  - Often referred to as Bayesian structure learning.

- Constraint-based method
  - Determine the conditional independence of the underlying probability by statistical tests.
  - Problem: Many statistical tests are required.
  - Often referred to as causal learning.

Score-based Structure Learning: Example

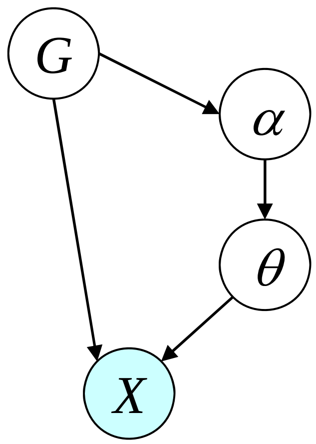
Discrete variables: $X_1, \ldots, X_m$

Data: $D = \{(X_1^{(1)}, \ldots, X_m^{(1)}), \ldots, (X_1^{(N)}, \ldots, X_m^{(N)})\}$

- Model:

  When a directed graph G is specified, multinomial distribution is assumed with Dirichlet prior.

$$p(X \mid \theta) = \prod_{b=1}^{m} p(X_b \mid X_{pa(b)}, \theta_b)$$



$\theta_b = (\theta_{b,i}^j)$ $\qquad i$: multi-index for $pa(b)$

$$\theta_{b,i}^j = P(X_b = j \mid X_{pa(b)} = i) \quad \theta_{b,i}^j \geq 0, \sum_{j=1}^{K_b} \theta_{b,i}^j = 1.$$

$$p(D \mid \theta) = \prod_{n=1}^{N} \prod_{b=1}^{m} p(X_b^{(n)} \mid X_{pa(b)}^{(n)}, \theta_b)$$

Dirichlet prior:

$$\theta_{b,i} = (\theta_{b,i}^1, \ldots, \theta_{b,i}^{K_b}) \sim \mathrm{Dir}(\theta_{b,i} \mid \alpha_{b,i}^1, \ldots, \alpha_{b,i}^{K_b}) = \frac{\Gamma(\sum_j \alpha_{b,i}^j)}{\prod_j \Gamma(\alpha_{b,i}^j)} \prod_{j=1}^{K_b} (\theta_{b,i}^j)^{\alpha_{b,i}^j - 1}$$

Score-based Structure Learning: Example

- Marginal likelihood:

Score(G) $\equiv$ Log Marginal Likelihood of G.

$$= \log \int P(D \mid \theta, G)\, p(\theta \mid G, \alpha)\, d\theta \qquad\qquad \alpha = (\alpha_{b,i}^{j})$$

$$= \log \int \prod_{b=1}^{m} \prod_{i=1}^{\#pa(b)} \prod_{j=1}^{K_b} (\theta_{b,i}^{j})^{N_{b,i}^{j}} \frac{\Gamma(\sum_j \alpha_{b,i}^{j})}{\prod_j \Gamma(\alpha_{b,i}^{j})} \prod_{j=1}^{K_b} (\theta_{b,i}^{j})^{\alpha_{b,i}^{j}-1} d\theta_{b,i}$$

$$= \sum_{b=1}^{m} \sum_{i=1}^{\#pa(b)} \left[\log\Gamma(\sum_j \alpha_{b,i}^{j}) - \sum_{j=1}^{K_b} \log\Gamma(\alpha_{b,i}^{j}) \right.$$

$$\left. - \log\Gamma(\sum_j \tilde{\alpha}_{b,i}^{j}) + \sum_{j=1}^{K_b} \log\Gamma(\tilde{\alpha}_{b,i}^{j}) \right]$$

where $\tilde{\alpha}_{b,i}^{j} = N_{b,i}^{j} + \alpha_{b,i}^{j}$

$N_{b,i}^{j}$: number of data s.t. $X_b = j$ and $X_{pa(b)} = i$.

Score-based Structure Learning

- Prior to the models

  We can use a prior distribution $P(G)$ on the graphs.

  $$\text{Score}(G) = \log P(D \mid G) + \log P(G)$$

- Optimization over the graphs

  The space is very huge → greedy search.

  Start from a graph G, and repeat the following process:

  Update the graph by deleting, inserting, or reversing an edge.

  Accept the new graph G' if Score(G') > Score(G).

- Many others
  - Score by MDL (minimum description length) / BIC (Bayesian information criterion)
  - MCMC, etc.

  See D. Heckerman "A tutorial on learning with Bayesian networks" in *Learning in Graphical Models* (M. Jordan ed. 1998).

Marginal Likelihood / ABIC

- Bayesian method for model selection

Maximum a posteriori model given data

$$\hat{G} = \arg\max P(G \mid D)$$

Note:

$$P(G \mid D) = \frac{P(D \mid G)P(G)}{P(D)} \propto P(D \mid G)P(G)$$ as a function of model

$$\Rightarrow \quad \hat{G} = \arg\max\left[\log P(D \mid G) + \log P(G)\right]$$

If $P(G)$ is uniform over the models,

$$\hat{G} = \arg\max \log P(D \mid G)$$

$$= \arg\max \log \int P(D \mid \theta, G) P(\theta \mid G) d\theta$$

Marginal log likelihood

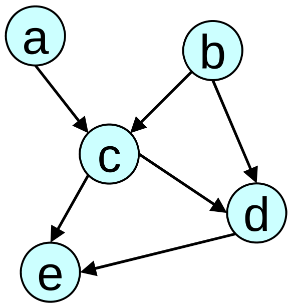
(ABIC: Akaike's Bayesian information criterion)

Mini-Summary on score-based method

- Use a global score to match a graph and data.
  Marginal log likelihood (ABIC), MDL, etc.
- Optimization in huge search space.
  Some techniques are needed. e.g. greedy search.
- Able to use informative prior on graphs.
- Usually, discrete or Gaussian variables are assumed.
  For non-Gaussian continuous variables, we need some techniques such as discretization.
- Also known as Bayesian structure learning

Causal Learning

- Directed graph as causal graph
  - A directed graph can be regarded as the expression of causal relationships among variables.



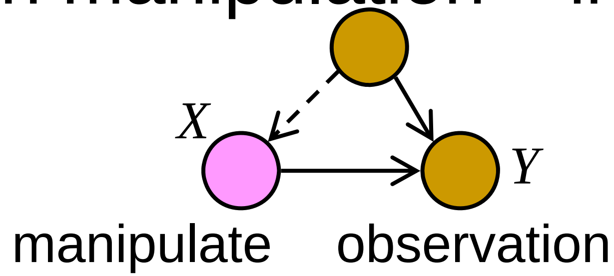
Causal direction = Edge-direction

$$p(X) = p(X_a)p(X_b)p(X_c \mid X_a, X_b) \\ \times p(X_d \mid X_b, X_c)p(X_e \mid X_c, X_d)$$

  - Causal learning: learning of the directed graph from data.

Causal Leaning from Data

- With manipulation – intervention



X is a cause of Y?

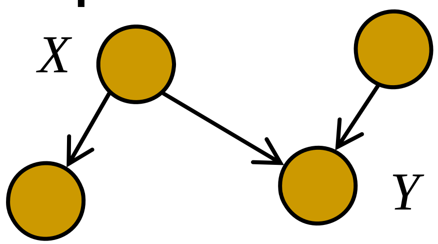
Easier. (*do*-calculus, Pearl 1995)

- No manipulation / with temporal information

$X(t) \quad Y(t)$: observed time series

$X(1), \ldots, X(t)$ are a cause of $Y(t+1)$?

- No manipulation / no temporal information



Causal inference is harder.

Addendum: Causality and Correlation

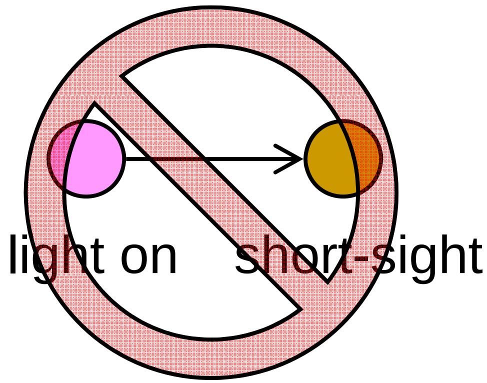
- Correlation (dependence) and causality

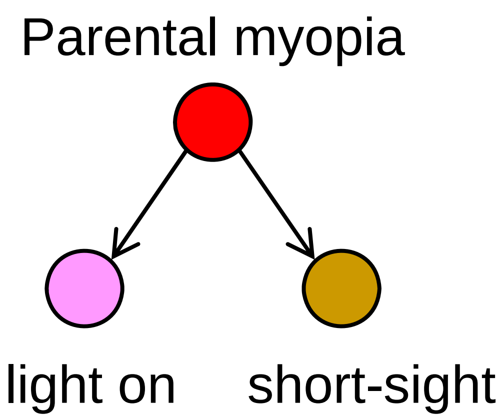
  Do not confuse causality with dependence (*or correlation*)!

  Example)

  A study shows:

  Young children who sleep with the light on are much more likely to develop myopia in later life. (*Nature* 1999)





(*Nature* 2000)

Hidden common cause

Causal Learning without Manipulation

- Difficulty of causal inference from non-experimental data
  - Widely accepted view till 80's

    Causal inference is impossible without manipulating some variables.

    e.g.) *"No causation without manipulation"* (Holland 1986, JASA)

  - Temporal information is very helpful, but not decisive.

    e.g.) The barometer falls before it rains, but it does not cause the rain.

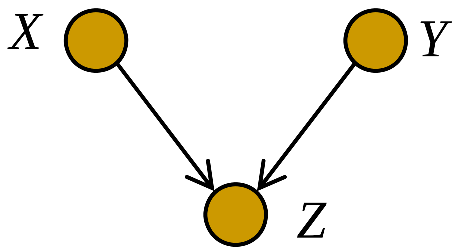
  - Many philosophical discussions, but not discussed here.

    See Pearl (2009) and the references therein.

Causal Learning without Manipulation

- Why is it possible?
  - DAG of chain X – Z – Y

V-structure



$$\Longleftrightarrow$$

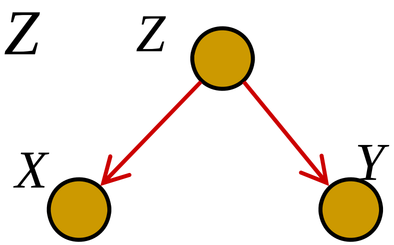
$$X \perp\!\!\!\perp Y$$

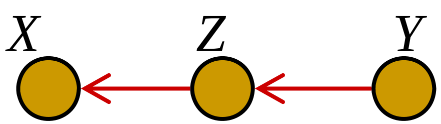
and

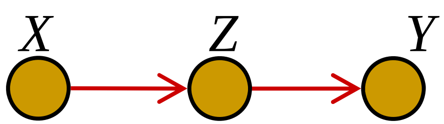
$$X \not\perp\!\!\!\perp Y \mid Z$$

  - This is the only detectable directed graph of three variables.
  - The following structures cannot be distinguished from the probability.

$$X \perp\!\!\!\perp Y \mid Z$$







p(x,y,z) = p(x|z)p(y|z)p(z) = p(x|z)p(z|y)p(y) = p(x|z)p(z|y)p(x)

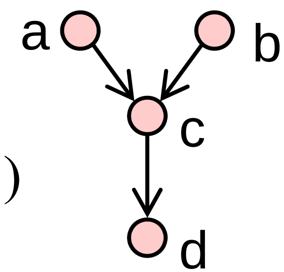
Causal Learning without Manipulation

- Fundamental assumptions
  - Causal Markov condition

    The probability generating data is associated with a DAG.

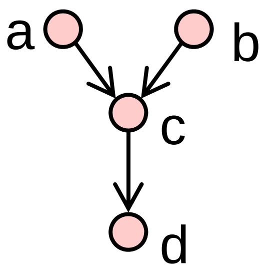
$$p(X) = \prod_{i=1}^{n} p(X_i \mid \mathrm{pa}(i))$$

$$p(X) = p(X_a)\, p(X_b)\, p(X_c \mid X_a, X_b)\, p(X_d \mid X_c)$$

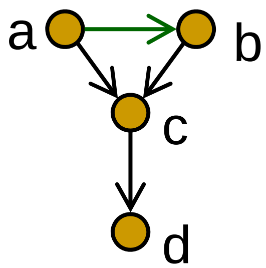


  - Causal Faithfulness Condition

    The inferred DAG (causal structure) must express all the independence relations.



true



unfaithful

This includes the true probability as a special case, but the structure does not express $a \perp\!\!\!\perp b$

Constraint-based Causal Learning

- IC algorithm (Verma&Pearl 90)

  Input – V: set of variables, D: dataset of the variables.

  Output – Partial DAG (specifies an equivalence class, directed partially)

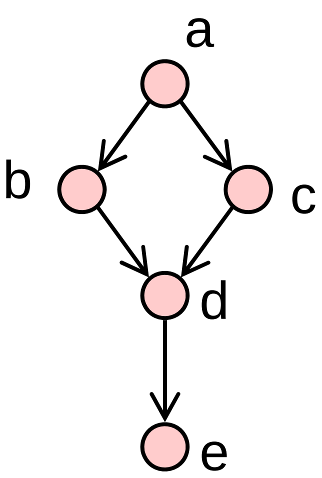
  1. For each $(a,b) \in V \times V \ (a \neq b)$, search for $S_{ab} \subset V \setminus \{a,b\}$ such that
     $$X_a \perp\!\!\!\perp X_b \mid S_{ab}$$
     Construct an undirected graph (skeleton) by making an edge between a and b if and only if no set S_{ab} can be found.
  2. For each nonadjacent pair (a,b) with $a - c - b$, direct the edges by $a \rightarrow c \leftarrow b$ if $c \notin S_{ab}$
  3. Orient as many of undirected edges as possible on condition that neither new v-structures nor directed cycles are created.

  – Implemented in PC algorithm (Spirtes & Glymour) efficiently.

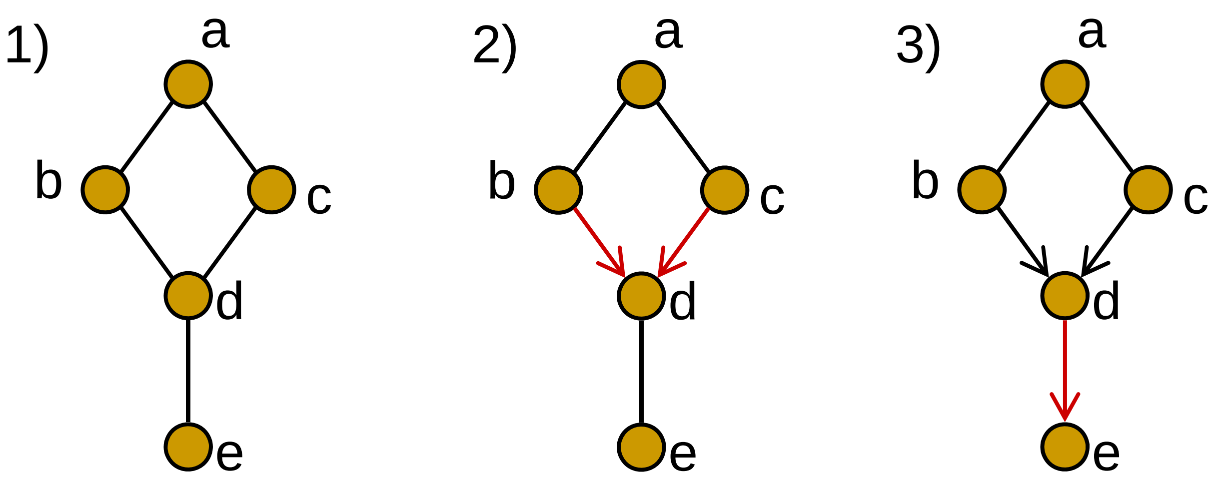
Constraint-based Causal Learning

- Example

True structure



The output from each step of IC algorithm



$$S_{ad} = \{b, c\}$$
$$S_{ae} = \{d\}$$
$$S_{bc} = \{a\}$$
$$S_{be} = S_{ce} = \{d\}$$

For other pairs,
S does not exist.

For (b,c), $d \notin S_{bc}$

Direction of some edges may be left undetermined.

Mini-summary on constraint-based method

- Determine the conditional independence of the underlying probability by statistical tests.
- Many statistical tests are required.
  - Problems:
    - Errors in statistical tests.
    - Computational costs.
    - Multiple comparison – difficult to set critical regions
- Effects of hidden variables are important to consider (not discussed here).
- Often discussed in the context of causal learning.

Summary: Structure learning

- Two major approaches
  - Score-based Bayesian structure learning

    There are many methods how to define score function.

    Marginal likelihood, MDL, etc.
  - Constraint-based causal learning

    Testing conditional independence.

- More recent approach
  - Sparse network by Lasso

    Meinshausen and Buhlmann [*Ann. Statist.* **34** (2006) 1436–1462]

- Further readings

  D. Heckerman. A tutorial on learning with Bayesian networks. in *Learning in Graphical Models*. (ed. M.Jordan) pp.301-354. MIT Press (1999)

  This book contains various advanced topics.

  J. Pearl. *Causality*. 2nd ed. Cambridge University Press (2009)

  宮川雅巳 「統計的因果推論」朝倉書店(2004)

  宮川雅巳 「グラフィカルモデリング」朝倉書店(1997)