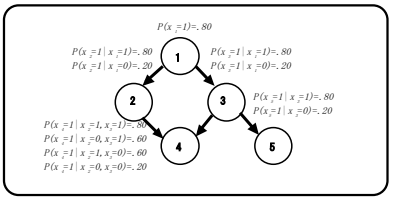
ベイジアン・ネットワークの学習

電気通信大学
情報理工学研究科
植野　真臣

ベイジアン・ネットワークモデル



$$P(x_1, x_2, \cdots, x_N \mid G) = \prod_{i=1}^{N} p(x_i \mid \Pi_i, G)$$

$\Pi_i \subseteq \{x_1, x_2, \cdots, x_{q_i}\}$　は変数iの親ノード集合

パラメータ化とパラメータ推定

ベイジアン・ネットワークのParametrerization
（Spiegelhalter, D.J.,and Lauritzen, S.L. 1990）

Spiegelhalter, D.J.,and Lauritzen, S.L. Sequential updating of conditional Probabilities on directed graphical structures. Networks 20 (1990), 579-605

今、θ_{ijk} を親ノード変数集合 Π_i がj番目のパターンをとったときの

$x_i = k$

となる条件付確率を示すパラメータとする。
このとき、データXを得たときの尤度は、以下のとおりである。

$$L(\Theta_S \mid \mathbf{X}, G) \propto \prod_{k=0}^{r_i-1} \theta_{ijk}^{\,n_{ijk}}$$

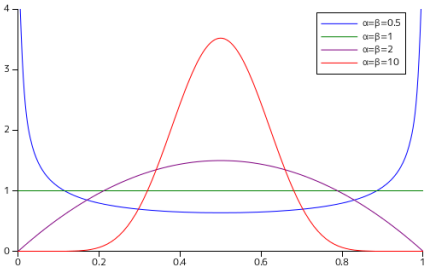
$$\propto \prod_{i=1}^{N} \prod_{j=1}^{q_i} \frac{\sum_{k=0}^{r_i-1} n_{ijk}!}{\prod_{k=0}^{r_i-1} n_{ijk}!} \prod_{k=0}^{r_i-1} \theta_{ijk}^{\,n_{ijk}}$$

Cooper and Herskovits, 1992 A Bayesian methods for the Induction of Probabilistic networks from data, Machine Learning, 9, 309-347

- 多項分布の自然共役分布である以下のディレクレイ分布を事前分布に導入

$$p(\Theta_S \mid G) = \prod_{i=1}^{N} \prod_{j=1}^{q_i} \frac{\Gamma(\sum_{k=0}^{r_i-1} \alpha_{ijk})}{\prod_{k=0}^{r_i-1} \Gamma(\alpha_{ijk})} \prod_{k=0}^{r_i-1} \theta_{ijk}^{\,\alpha_{iij}-1}$$

ディレクレィ分布の
周辺分布ベータ分布



事後分布

$$p(\mathbf{X},\Theta_G \mid G)=\prod_{i=1}^{N}\prod_{j=1}^{q_i}\frac{\Gamma(\sum_{k=0}^{r_i-1}(\alpha_{ijk}+n_{ijk}-1))}{\prod_{k=0}^{r_i-1}\Gamma(\alpha_{ijk}+n_{ijk}-1)}\prod_{k=0}^{r_i-1}\theta_{ijk}^{\ \alpha_{ijk}+n_{ijk}-1}$$
$$\propto\prod_{i=1}^{N}\prod_{j=1}^{q_i}\prod_{k=0}^{r_i-1}\theta_{ijk}^{\ \alpha_{ijk}+n_{ijk}-1}$$

MAP推定量

$$\hat{\theta}_{ijk}=\frac{\alpha_{ijk}+n_{ijk}-1}{\alpha_{ij}+n_{ij}-r_i}$$

ただし　$\alpha_{ij}=\sum_{k=0}^{r_i-1}\alpha_{ijk}$　$n_{ij}=\sum_{k=0}^{r_i-1}n_{ijk}$

EAP推定量

$$\hat{\theta}_{ijk}=\frac{\alpha_{ijk}+n_{ijk}}{\alpha_{ij}+n_{ij}}$$

ただし　$\alpha_{ij}=\sum_{k=0}^{r_i-1}\alpha_{ijk}$　$n_{ij}=\sum_{k=0}^{r_i-1}n_{ijk}$

表1

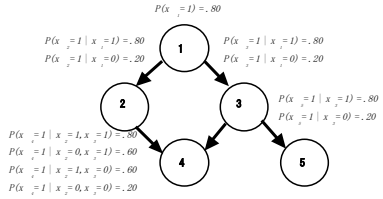
	ノード番号				
	1	2	3	4	5
1	1	0	1	1	1
2	0	0	0	1	0
3	1	1	1	1	1
4	1	1	1	1	0
5	1	1	1	1	0
6	1	1	0	0	0
7	1	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	1	1	1	1
11	0	0	0	0	0
12	1	1	1	1	1
13	0	0	1	1	1
14	1	1	1	1	1
15	1	0	1	1	1
16	1	1	1	0	0
17	1	1	0	1	0
18	1	1	1	0	0
19	1	1	0	1	0
20	1	1	1	0	1
平均	0.70	0.60	0.60	0.60	0.40

表1より推定した母数推定値

	真の値	$\alpha_{ijk}=0$(最尤推定)	$\alpha_{ijk}=1$	$\alpha_{ijk}=1/2$
$P(x_2=1\|x_1=1)$	0.8	0.8.	0.76	0.78
$P(x_2=1\|x_1=0)$	0.2	0	0.14	0.08
$P(x_3=1\|x_1=1)$	0.8	0.8	0.76	0.78
$P(x_3=1\|x_1=0)$	0.2	0	0.14	0.08
$P(x_4=1\|x_2=1, x_3=1)$	0.8	0.66	0.64	0.65
$P(x_4=1\|x_2=1, x_3=0)$	0.6	0.5	0.5	0.5
$P(x_4=1\|x_2=0, x_3=1)$	0.6	0.5	0.5	0.5
$P(x_4=1\|x_2=0, x_3=0)$	0.2	0.4	0.43	0.41
$P(x_5=1\|x_3=1)$	0.8	0.67	0.64	0.42
$P(x_5=1\|x_3=0)$	0.2	0.33	0.35	0.34
真の値との平均自乗誤差		0.0193	0.015	0.028

Learning Bayesian Networks

データから構造を推定する



モデル選択基準

もっとも良いモデルをデータから選択する基準

以下を最小化するモデルを選べばよい

Akaike Information Criterion

AIC = - 2 ln-Likelihood + 2 No.Parameters

Bayesian Information Criterion

BIC = -2 ln-Likelihood + No.Parameters ln(n)

ここで　nはデータ数

AICは期待対数尤度の近似，BICは周辺尤度の近似

AICは漸近的一致性を持たないが，BICは持つ．

AICとBICの条件

- AICとBICは，統計的正則モデルを仮定している。
- 統計的正則モデル：最尤推定量が正規分布に法則収束するモデルを 正則モデルと呼ぶ。このとき，漸近的にフィッシャー情報量行列の固有値がすべて0よりも大きいので漸近展開により，AICやBICが導出される。

ベイジアンネットワークは

- 統計的正則性を持たない。
- 情報科学で用いられる数理モデルは，ほとんど統計的正則性を持たない。
- 理論的には　AICやBICを用いることは問題がある。

ディレクレィ分布の周辺尤度を直接計算

$$p(G \mid X) \propto P(G) \int_{\Theta_S} p(\mathbf{X}_s \mid \Theta_G G) p(\Theta_G) d\Theta_G$$

$$= P(G) \prod_{i=1}^{N} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ijk})}{\Gamma\left[\sum_{k=0}^{r_i-1} (\alpha_{ijk} + n_{ijk})\right]} \prod_{k=0}^{r_i-1} \frac{\Gamma(\alpha_{ijk} + n_{ijk})}{\Gamma(\alpha_{ijk})}$$

$$= P(G) \prod_{i=1}^{N} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ijk})}{\Gamma(\alpha_{ij} + n_{ij})} \prod_{k=0}^{r_i-1} \frac{\Gamma(\alpha_{ijk} + n_{ijk})}{\Gamma(\alpha_{ijk})}$$

K2
(Cooper and Herskovits 1992)
α_{ijk}=1（一様分布）のとき

$$p(G \mid \mathbf{X}) \propto p(G) \prod_{i=1}^{N} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(n_{ij} + r_i - 1)!} \prod_{k=0}^{r_i-1} n_{ijk}!$$

D.Heckerman, D.Geiger and D.M.Chickering (1995)

- Likelihood equivalence

G_1 G_2 の構造がマルコフ確率構造として同型であるとき

$$p(\Theta_U \mid G_1) = p(\Theta_U \mid G_2)$$

が成り立つこと

$\alpha_{ijk} = 1.0$ のときの周辺尤度

今、二つの変数 x, y について二つの背反するLikelihood equivalenceの構造 $G_{x \to y}$ $G_{y \to x}$ を考える。

$$p(G \mid \mathbf{X}) \propto p(G) \prod_{i=1}^{N} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(n_{ij} + r_i - 1)!} \prod_{k=0}^{r_i - 1} n_{ijk}!$$

を用いた場合、

$$p(G_{x \to y} \mid X) \neq p(G_{y \to x} \mid X)$$

となり、Likelihood equivalenceの仮定を満たさない。

BDe Score Metric

(Likelihood equivalent Bayesian Dirichlet scoring)

- Likelihood equivalenceの仮定を満たすScoring Metricの十分条件は

$$P(G) \prod_{i=1}^{N} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ijk})}{\Gamma(\alpha_{ij} + n_{ij})} \prod_{k=0}^{r_i - 1} \frac{\Gamma(\alpha_{ijk} + n_{ijk})}{\Gamma(\alpha_{ijk})}$$

ただし $\alpha_{ijk} = \alpha p(x_i = k, \Pi_i = j \mid G)$

α は事前分布の擬似サンプル数ESSで事前分布の重みでもある

BDeu Score Metric (W.L.Buntine (1991))

- Bdeの一様分布を考えたモデル

$$P(G) \prod_{i=1}^{N} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ijk})}{\Gamma(\alpha_{ij} + n_{ij})} \prod_{k=0}^{r_i - 1} \frac{\Gamma(\alpha_{ijk} + n_{ijk})}{\Gamma(\alpha_{ijk})}$$

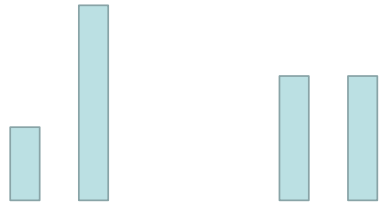
ただし

$$\alpha_{ijk} = \alpha / (q_i r_i)$$

問：ハイパーパラメータαを大きくするとエッジは付きやすくなるのか？

$$\hat{\theta}_{ijk} = \frac{\alpha_{ijk} + n_{ijk}}{\alpha_{ij} + n_{ij}}$$

ただし $\alpha_{ij} = \sum_{k=0}^{r_i - 1} \alpha_{ijk}$ $n_{ij} = \sum_{k=0}^{r_i - 1} n_{ijk}$



P(X=1|Y=1) P(X=1|Y=0)
真の条件付き確率

P(X=1|Y=1) P(X=1|Y=0)
ハイパーパラメータαを大きくするとその推定値は一様分布に近づく

予測

- ESSを大きくするとエッジはつきにくくなる

Ueno2010

$$logp(x|\alpha, G) = \sum_{i=1}^{N}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}(\alpha_{ijk}+n_{ijk})log\frac{\alpha_{ijk}+n_{ijk}}{\alpha_{ij}+n_{ij}} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}\frac{r_i-1}{r_i}\log\left(1+\frac{n_{ijk}}{\alpha_{ijk}}\right)+ \boldsymbol{O}(1)$$

定理36 (AICの事前知識表現：Ueno(2010))

$For\ \forall i,\forall j,\forall k, \alpha_{ijk}=\frac{1}{3}n_{ijk}$ のとき，対数周辺尤度スコアはAICにより $\boldsymbol{O}(1)$ で近似される.

AICではあらかじめ真のモデルがわかっているのと同じ

定理37 (BICの事前知識表現：Ueno(2010))

$For\ \forall i,\forall j,\forall k, \alpha_{ijk}=1.0$ (事前分布が一様分布，K2(Cooper and Herskovits 1992)に一致)のとき，対数周辺尤度スコアはBICにより $\boldsymbol{O}(1)$ で近似される.

系2（ $\log-$ BDeuのESSによるトレードオフ：Ueno(2010))

$\alpha+n$が十分大きいとき，$\log-$BDeuは以下に近似できる.

$$\log p(x|G) = \alpha\sum_{i=1}^{N}\log r_i + \sum_{i=1}^{N}\sum_{j=1}^{q_i}\sum_{k=0}^{r_i-1}\left(\frac{\alpha}{r_i q_i}+n_{ijk}\right)\log\frac{\frac{\alpha}{r_i q_i}+n_{ijk}}{\frac{\alpha}{q_i}+n_{ij}} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{q_i}\sum_{k=0}^{r_i-1}\frac{r_i-1}{r_i}\log\left(1+\frac{r_i q_i n_{ijk}}{\alpha}\right)$$

定理38（ESS増大による完全グラフ生成：Ueno(2010))

ESSを大きくしていくと，学習されたベイジアンネットワーク構造のエッジ数は担当増加し，完全グラフ（complete graph）に近づいていく.

定理39（ESS減少によるからグラフ生成：Ueno(2010))

ESSを小さくしていくと，学習されたベイジアンネットワーク構造のエッジ数が単調減少し，空グラフ（empty graph）に漸近的に近づいていく.

データへのESS値の最適性

- 真の条件付き確率分布が非一様のとき：
  最適なESS値は比較的小さい値を与えなければならない．なぜならば，式(6.24)右辺において，対数事後分布はエッジを追加しようとするので過学習を起こしやすい．第2項で過学習を抑えるためにはESS値は小さい値に設定しなければならない.
- 真の条件付き確率分布がほぼ一様のとき：
  最適なESS値は比較的大きな値を与えなければならない．なぜならば，真の条件付き確率分布が一様なので式(6.24)右辺において，対数事後分布は真の因果を発見することが難しい．第2項で過小学習（underfitting）を避けるためにESS値は大きい値に設定しなければならない.
- スパース（過疎）なデータのとき：
  n_{ijk}が欠測データもしくは過疎である場合，なるべくESSの影響を抑え，データの学習への影響を最大にしなければならない．式(6.24)より$\mathrm{ESS}=1.0$となるとき，データの影響を最大化できる．Castillo Hadi and Solares(1997)　は周辺尤度の分散がn_{ijk}が欠測のとき，ESSの2乗に反比例することを示している．すなわち，この意味でも$\mathrm{ESS}=1.0$がデータの影響を最大にできることがわかる.

定理40(ESS減少による空グラフ生成：Ueno(2011))

$\alpha < r_i q_i, (i = 1, \dots, N)$, nが十分大きいとき，対数周辺尤度は以下に収束する.

$$\log p(x|g,\alpha) = \sum_{i=1}^{N}\sum_{j=1}^{q_i}\sum_{k=0}^{r_i-1}\left(\frac{\alpha}{r_i q_i}+n_{ijk}\right)\log\frac{\frac{\alpha}{r_i q_i}+n_{ijk}}{\frac{\alpha}{q_i}+n_{ij}} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{q_i}\left[\frac{r_i-1}{r_i}\sum_{k=1}^{r_i}\log\left(\frac{(r_i q_i)^2 n_{ijk}}{2\pi\alpha^2}\right)\right] + \boldsymbol{O}(1)$$

定理41（最適なESSの決定：Ueno(2010))

BDe(u)は，$\mathrm{ESS} = 1.0$のとき，事後分布の分散を最大化する.

いま最もよいスコア 事前知識に頑健な基準

定義87($\mathrm{NIP-BIC}$：Ueno(2011))

$$\mathrm{NIP-BIC} = \sum_{i=1}^{N}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}(\alpha_{ijk}+n_{ijk})\log\frac{\alpha_{ijk}+n_{ijk}}{\alpha_{ij}+n_{ij}} - \frac{1}{2}q_i r_i \log(1+n)$$

完全無情報事前分布

定義88(完全無情報事前分布スコア$\mathrm{NIP-BDe}$：

$$p(x|g,\alpha) = \sum_{g^h\in G} p(g^h)p(x|\alpha,g,g^h) = \sum_{g^h\in G} p(g^h)\left(\prod_{i=1}^{N}\prod_{j=1}^{q_i}\frac{\Gamma\left(\alpha_{ij}^{g^h}\right)}{\Gamma\left(\alpha_{ij}^{g^h}+n_{ij}\right)}\prod_{k=1}^{r_i}\frac{\Gamma\left(\alpha_{ijk}^{g^h}+n_{ijk}\right)}{\Gamma\left(\alpha_{ijk}^{g^h}\right)}\right)$$

ここで$\Sigma_{g^h\in G}$はすべての可能な構造候補に対する和，$p(g^h)$は構造候補の事前分布でここでは一様分布を仮定している.

構造の探索アルゴリズム

変数の数nに対して、構造の候補数は以下のように増える。

$$f(n) = \sum_{i=1}^{n}(-1)^{i+1}\begin{bmatrix} n \\ i \end{bmatrix} 2^{i(n-i)} f(n-i)$$

例えば、n=2のとき、構造数は3、n=3のとき、構造数は25、n=5で29000,n=10で4.2×10^{18}

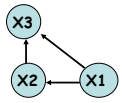
探索問題は、指数爆発する。。なんらかの工夫が必要。

アルゴリズム

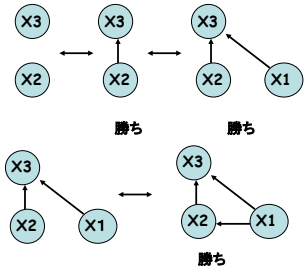
1. 変数間の順序を決める。X1>x2>・・>xn
2. Xnのすべての親ノードパターンを変えながら、情報量基準でどのパターンが最適かを検索。親ノードパターンは、親ノード数をm個に制限するのが普通である。
3. Xn-1について、1と同じ手続きを行い、X1まで繰り返す。
4. 最もよい値を置いておく
5. すべての順序について１－４を繰り返す

例

- X1>X2>x3から始める



真の構造

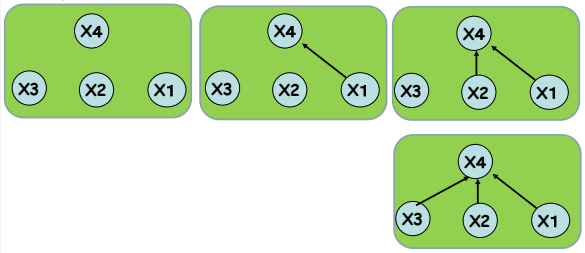


- X1>X3>X2
- X2>X3>X1
- X2>X1>X3
- X3>X2>X1
- X3>X1>X2

について同様のことを行い，最もスコアの高かった構造を推定値とする

親探しアルゴリズムが重要

- 順序を所与として，
- 変数Xiの親ノード候補から、いかに早く親ノードをみつけてくるかのためのアルゴリズム
- 例　X4>X3>X2>X1



親ノード探索に用いられるアルゴリズム

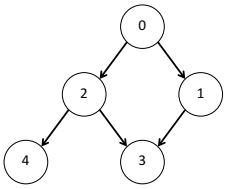
- 動的計画法DP　O(N2^N)
- A＊探索
- 幅優先探索と分枝限定法（BFBnB）

動的計画法による構造学習アルゴリズム

- Tomi Silander and Petri Myllymaki. A simple approach for finding the globally optimal bayesian network structure. In Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06), Arlington, Virginia, 2006. AUAI Press.

準備

- 変数集合:$V = \{V_1, \cdots, V_n\} = \{1, \cdots, n\}$
- ネットワーク：$G = \{G_1, \cdots, G_n\}(G_i$ は V の部分集合かつ V_i の親ノード集合)
- 変数順序：例えば，$ord(4,3,2,1)$
- 変数順序の i 番目の要素：$ord_2 = 3$
- トポロジー順序：G のすべての要素 G_i は，$G_i \subseteq \cup_{j=1}^{i-1}\{ord_j\}$ を満たす
- "$sinks$"：外向きのアークを持たず，他のどのノードの親にならないノード



$V = \{0,1,2,3,4\}$

Order =(0,2,4,1,3)

G ={{},{0},{2},{0},{1,2}}

sinks = {3,4}

動的計画法による構造推定アルゴリズム

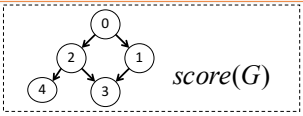
1. 全ての（変数、親変数集合）のペアに対してローカルスコアを計算
2. 各変数について、すべての親変数集合パタンに対する最適な親変数集合を決定
3. 全ての変数集合パタン$W \subset V$に対して、Sinkをひとつ選択
4. ステップ3の結果を用いて、最適な変数順序を決定
5. ステップ2と4の結果を用いて、最適なネットワーク構造を決定

ローカルスコア　関数　$CFT(i, G_j)$に関する同時確率分布表作成

最適な構造：スコア関数(BIC, BDeu,…)を最大化

ネットワーク全体のスコア
＝各ノードとその親ノード集合のみで構成される局所的な構造のスコア
（ローカルスコア）

$$score(G) = \sum_{i=1}^{n} score(G_i) = \sum_{i=1}^{n} score(CFT(i, G_i))$$

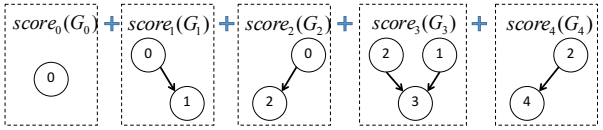


$$BIC = \log(Likelihood) - \frac{K}{2}\log(n)$$

$$BDeu = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(n'_{ijk})}{\Gamma(n'_{ij}+n_{ij})}\prod_{k=0}^{r_i}\frac{\Gamma(n'_{ijk}+n_{ijk})}{\Gamma(n'_{ijk})}$$

$$N'_{ijk} = N'/(q_i r_i)$$

$score_0(G_0) + score_1(G_1) + score_2(G_2) + score_3(G_3) + score_4(G_4)$



手順1：全ての（変数，親変数集合）の組み合わせに対してローカルスコアを計算

Algorithm 1 $GetLocalScores(ct, evars)$

```
/* Turn ct to cfts and further to local scores */
for all v ∈ Vars(ct) do
    LS[v][Vars(ct) \ {v}] ← Score(Ct2cft(ct, v))
end for

/* Recursively call GetLocalScores */
if |Vars(ct)| > 1 then
    for all v ∈ evars do
        GetLocalScores(Ct2ct(ct, v), {1, ··· , v − 1})
    end for
end if
```

関数Ct2ct(ct,v)
同時確率分布表ctから変数vを周辺化した同時確率分布表を生成する関数

②

①

Child=0	W={1} -528	W={2} -520	W={1,2} -522	W={3} -526	W={1,3} -515	W={2,3} -510	W={1,2,3} -501	…
Child=1	W={0} -535	W={2} -524	W={0,2} -528	W={3} -530	W={0,3} -531	W={2,3} -524	W={0,2,3} -521	…
Child=2	W={0} -537	W={1} -520	…					…

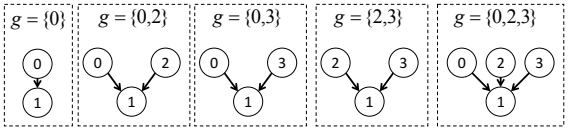
手順2最適な親ノード集合：Best Parent Set：BPS

各変数iに対して，任意の変数集合Cを所与としたときの最適な親ノード集合：

$$g_i^*(C) = \arg\max_{g \subseteq C} \{score_i(g)\}$$

例 $g_1^*(C = \{0,2,3\}) = \arg\max_{g \subseteq C} \{score_1(g)\}$ ＝以下の中でも最もローカルスコアが高い親ノード集合

$score_1(\{0\})$ $score_1(\{0,2\})$ $score_1(\{0,3\})$ $score_1(\{2,3\})$ $score_1(\{0,2,3\})$



手順2：最適な親ノード集合BPSの探索

Algorithm 2 $GetBestParents(V, v, LS)$

```
bps = array 1 to 2^(|V|−1) of variable sets
bss = array 1 to 2^(|V|−1) of local scores
for all cs ⊆ V \ {v} in lexicographic order do
    bps[cs] ← cs
    bss[cs] ← LS[v][cs]
    for all cs1 ⊂ cs such that |cs \ cs1| = 1 do
        if bss[cs1] > bss[cs] then
            bss[cs] ← bss[cs1]
            bps[cs] ← bps[cs1]
        end if
    end for
end for
return bps
```

ここがキモ：動的計画法

計算順序をLexicographic Order（辞書式順序）にすることで前の計算結果を再利用できる

辞書式順序

ビット表現　====> {3,2,1}の1が立っている変数集合を候補とする

- 001　===> {1}
- 010　===> {2}
- 011　===> {1,2}
- 100　===> {3}
- 101　===> {1,3}
- 110　===> {2,3}
- 111　===> {1,2,3}

手順2の例

ノード集合{0,1,2,3}のとき、ノード0を子ノードとする場合を考える

ローカルスコアが次のように得られているとする

Child=0	Π_0={1} -528	Π_0={2} -520	Π_0={1,2} -522	Π_0={3} -526	Π_0={1,3} -515	Π_0={2,3} -510	Π_0={1,2,3} -521

このとき{1,2,3}のすべての部分集合Wについて、最適親ノード集合BPSとBPSのときのスコア**Best Score Set (BSS)**は以下のようになる

Child=0	W={1} bss -528 bps {1}	W={2} bss -520 bps {2}	W={1,2} bss -520 bps {2}	W={3} bss -526 bps {3}	W={1,3} bss -515 bps {1,3}	W={2,3} bss -510 bps {2,3}	W={1,2,3} bss -510 bps {2,3}

計算は左から順に行う（辞書式順序）

W={1,2,3}のBPSは、W={1,2}、W={1,3}、W={2,3}のBSSのみ比較し、最大のときのBPSとすればよい

手順3　最適なネットワークの探索

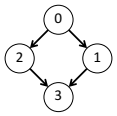
BPSを利用することで、最適な変数順序がわかっていれば、最適なネットワークが定まる

例：変数順序が{0, 1, 2, 3}の場合、以下が最適グラフ

$bps[0][\{\phi\}], bps[1][\{0\}], bps[2][\{0,1\}], bps[3][\{0,1,2\}]$

$$bps[0][\{\phi\}] = \{\phi\}, bps[1][\{0\}] = \{0\},$$
$$bps[2][\{0,1\}] = \{0\}, bps[3][\{0,1,2\}] = \{1,2\}$$

なら最適構造は右図となる

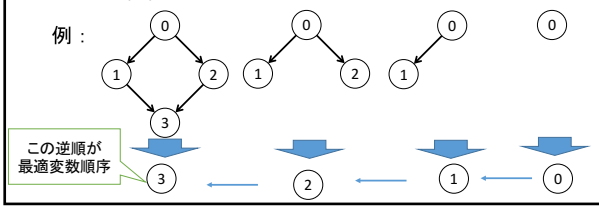


ただし、
得られる構造は変数順序に依存する

最適な変数順序の決め方

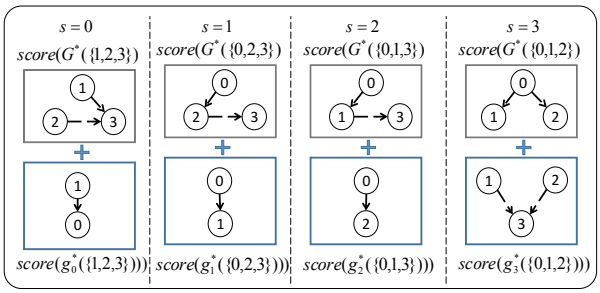
1. 全変数集合Vを所与としたとき，Sinkノード $sink(V)$ をひとつ選ぶ
2. 変数集合$W = V \setminus sink(V)$を所与としたとき，　Sinkノード $sink(W)$をひとつ選ぶ
3. 全変数集合$W' = W \setminus sink(W)$を所与としたとき，Sinkノード $sink(W')$をひとつ選ぶ
4. 3. を|W|=1まで繰り返す．

例：



最適なSinkの決め方

最適なSinkノード＝SinkとそのBPSのローカルスコア
＋Sink以外のノードで構成される最適構造のスコア
が最大となる



Sink決定の手順　（ここも動的計画法）

ここでも辞書式順序を使い、同じ計算を回避する

{ϕ}　Sinkなし
{1} 最適Sink=1, 集合{1}で構成される最適構造のスコア= LocalScore(1),
{2} 最適Sink=2, 集合{2}で構成される最適構造のスコア= LocalScore (2),
{3} 同上
{4} 同上
{1,2} 最適Sinkは, 以下の比較で決定
　　BSS(1 <- 2)+集合{2}で構成される最適構造のスコア（計算済み）
　　BSS(2 <- 1)+集合{1}で構成される最適構造のスコア（計算済み）
　　上が高ければSink=1、そうでなければSink=2
　高い方のスコアを集合{1,2}で構成される最適構造のスコアとして保持しておく
{1,3} 同上
{1,4} 同上
{2,3} 同上
{2,4} 同上
{3,4} 同上

Sink決定の手順 続き

{1,2,3} 最適Sinkは, 以下の比較で決定
　BSS(1 <- {2,3})+集合{2,3}で構成される最適構造のスコア（計算済み）
　BSS(2 <- {1,3})+集合{1,3}で構成される最適構造のスコア（計算済み）
　BSS(3 <- {2,3})+集合{2,3}で構成される最適構造のスコア（計算済み）
　上が高ければSink=1、二番目が高ければSink=2、下が高ければSink=3
　一番高いスコアは集合{1,2,3}で構成される最適構造のスコアとして保持
{1,2,4} 同上
{1,3,4} 同上
{2,3,4} 同上
{1,2,3,4}最適Sinkは, 以下の比較で決定
　BSS(1 <- {2,3,4})+集合{2,3,4}で構成される最適構造のスコア（計算済み）
　BSS(2 <- {1,3,4})+集合{1,3,4}で構成される最適構造のスコア（計算済み）
　BSS(3 <- {1,2,4})+集合{1,2,4}で構成される最適構造のスコア（計算済み）
　BSS(4 <- {1,2,3})+集合{1,2,3}で構成される最適構造のスコア（計算済み）

Sink決定の疑似コード

Algorithm 3 $GetBestSinks(V, bps, LS)$

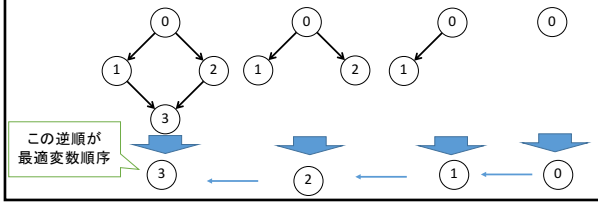
```
for all W ⊆ V in lexicographic order do
  scores[W] ← 0.0
  sinks[W] ← −1
  for all sink ∈ W do
    upvars ← W \ {sink}
    skore ← scores[upvars]
    skore ← skore + LS[sink][bps[sink][upvars]]
    if sinks[W] = −1 or skore > scores[W] then
      scores[W] ← skore
      sinks[W] ← sink
    end if
  end for
end for
return sinks
```

変数順序決定の疑似コード

Sinkセットを所与として変数順序を決定

Algorithm 4 $Sinks2ord(V, sinks)$

```
ord = array 1 to |V| of variables
left = V
for i = |V| to 1 do
  ord[i] ← sinks[left]
  left ← left \ {ord[i]}
end for
return ord
```



最適なネットワークの探索の疑似コード

Algorithm 5 $Ord2net(V, ord, bps)$

```
parents = array 1 to |V| of variable sets
predecs ← ∅
for i = 1 to |V| do
  parents[i] ← bps[ord[i]][predecs]
  predecs ← predecs ∪ {ord[i]}
end for
return parents
```

例：オーダー {0, 1, 2, 3} を所与として最適グラフ探索

$bps[0][\{\phi\}], bps[1][\{0\}], bps[2][\{0,1\}], bps[3][\{0,1,2\}]$

$bps[0][\{\phi\}] = \{\phi\}, bps[1][\{0\}] = \{0\},$
$bps[2][\{0,1\}] = \{0\}, bps[3][\{0,1,2\}] = \{1,2\}$

なら最適構造は右図となる

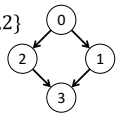


表 6.5　実験結果

データセット			計算時間〔秒〕			必要容量〔B（バイト）〕	
データセット	n	N	DP	BFBnB	A^*	DP	BFBnB
wine	14	178	1	0	0	1.16E+07	2.72E+05
houseVotes	17	435	7	5	3	4.81E+07	4.39E+06
hepatitis	20	126	27	9	6	3.79E+08	2.73E+07
segment	20	2 310	44	28	42	3.79E+08	3.67E+07
meta	22	528	52	57	41	1.67E+09	1.55E+08
import	22	205	123	54	55	1.67E+09	1.52E+08
parkinsons	23	195	297	103	130	3.48E+09	2.63E+08
sensorareadings	25	5 456	12 747	3 061	OM	1.51E+10	1.30E+09
autos	26	159	2 737	1 184	OM	3.15E+10	2.19E+09
flag	29	194	41 733	12 935	OM	2.81E+11	1.55E+10
wdbc	31	569	OD	93 682	OM	OD	6.86E+10
epgenetic	33	72 228	OD	570 760	OM	OD	2.74E+11

表 6.5 の OM はメモリオーバーで探索が打ち切られたことを表し，OD はハードディスクスペースに入らずに探索が打ち切られたことを表す．

厳密学習の問題点

■ 構造学習アプローチにおける探索の試行回数

- 5変数の場合: 80の試行
- 10変数の場合: 5120の試行
- 100変数の場合: **6.3383 x 10^{31}**の試行

計算量が
指数オーダー

■ これまでこの問題を解消するために
情報理論・コンピュータサイエンスの分野で提案されてきた手法

提案されてきた手法	最大変数数
動的計画法[Silander+06]	29
A*探索[Yuan et al., 2011]	24
幅優先分岐限定法[Malone et al., 2011]	33
整数計画法[Cussens, 2011]	**60**

未だ60変数が限界

制約ベースアプローチ

統計的因果モデルの分野
条件付き独立性(CI)テストと方向付けにより構造を
推定する手法が開発されている→**制約ベースアプローチ**

- 制約ベースアプローチのアルゴリズム

アルゴリズム	計算量
PC[Spirtes et al., 2000]	$O(N^k)$
MMPC[Tsumardinos et al.,2006]	$O(N2^{\|CPC\|})$
RAI[Yahezkel et al., 2009]	$O(N'^{k'})$

N: 変数数, k: 親変数数
N'<= N, k'<= k, CPC <= k

条件付き独立性（CI）テスト

制約ベースアプローチで用いられる独立検定手法

χ^2検定，G^2検定，条件付き相互情報量（CMI）

問題点

χ^2検定，G^2検定：
有意水準に精度依存，データ数に関係なく第一種の過誤（Type I error）が発生する．

CMI:
閾値に強く影響を受け，一致性を持たない．

真の構造を学習できる保証がない

本研究のアプローチ

ベイズ統計分野

- 統計的仮説検定手法に代わるBayes factorが提案されている[Kass et al., 1995].
- Bayes factorは2つのモデルの周辺尤度比により厳密なモデル選択が可能である．

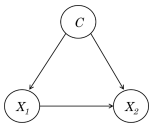
ベイジアンネットワーク

- Steckらにより，すでにBayes factorを用いたCIテストが提案されているが，構造学習の理論解析に用いられたにすぎず，構造学習に適用されていない．

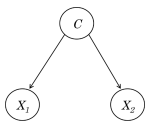
- 本研究では，Bayes factorを用いたCIテストを制約ベースアプローチの学習アルゴリズムに組み込む．
- 漸近一致性を持つ大規模構造学習を実現する．

Bayes factorを用いたCIテスト[Steck+02]

変数集合$\mathbf{C}$を所与としてX_1X_2間に辺がある構造をg_1，
辺がない構造をg_2とする．



g_1:辺がある構造



g_2:辺がない構造

観測されたデータ集合$\mathbf{D}$のとき，構造g_1とg_2の
対数Bayes factor: $\log\dfrac{p(\mathbf{D}\mid g_1)}{p(\mathbf{D}\mid g_2)}$

これを制約ベースアプローチの構造学習に適用する

提案手法[Natori+15]

- Steckらの手法では，X_1，X_2間を有向辺としているが，制約ベースアプローチでは無向グラフを学習する．
- 有向辺を仮定することで，各変数の親変数パラメータ数が異なり，CIテストの精度が安定しない．
- 各変数間の親変数パラメータを一定にする周辺尤度スコアを新たに提案する．
- 従属な構造のモデル

$$p(\mathbf{D}\mid g_1)=\prod_{j=1}^{q}\frac{\Gamma(r_1r_2\alpha_{g_1})}{\Gamma(r_1r_2\alpha_{g_1}+n_j)}\prod_{k_1=1}^{r_1}\prod_{k_2=1}^{r_2}\frac{\Gamma(\alpha_{g_1}+n_{jk_1k_2})}{\Gamma(\alpha_{g_1})}$$

- 独立な構造のモデル

$$p(\mathbf{D}\mid g_2)=\prod_{i=1}^{2}\prod_{j=1}^{q}\frac{\Gamma(r_i\alpha_{g_i})}{\Gamma(r_i\alpha_{g_i}+n_{ij})}\prod_{k_i=1}^{r_i}\frac{\Gamma(\alpha_{g_i}+n_{jk_i})}{\Gamma(\alpha_{g_i})}$$

本研究で用いたハイパーパラメータ：

$\frac{1}{2}$[Clarke et al., 1994], 1.0

RAIアルゴリズム

制約ベースアプローチにおいて最先端のアルゴリズム

学習手順

入力: データから生成される完全無向グラフ
出力: 学習により推定されたグラフ

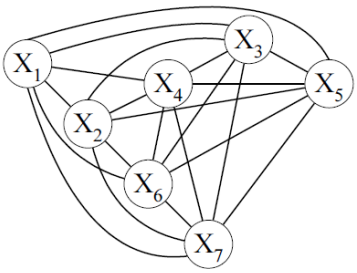
終了条件：各変数が次数n + 1より少ない潜在親変数数をもつ

1. グラフ上の各変数間の独立性をCIテストを用いて判定
2. 各辺を方向付け
3. 全体構造を部分構造に分割

終了条件を満たすまで手順1 ~ 3を再帰的に繰り返す

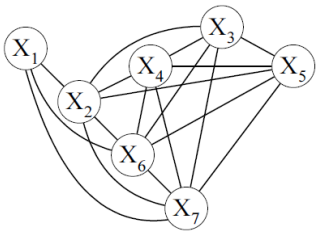
RAIアルゴリズム[Yehezkel +09]

- 入力は完全無向グラフ



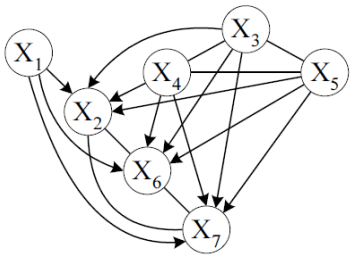
RAIアルゴリズム[Yehezkel +09]

1. CIテストの実行により不要な辺を削除



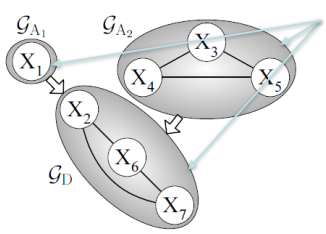
RAIアルゴリズム[Yehezkel +09]

2. 方向付けを行う



RAIアルゴリズム[Yehezkel +09]

3. 方向付けの結果から親変数集合と子変数集合に分割



分割した構造毎で1~3の処理を実行する

厳密学習との計算量の比較

- 厳密学習における計算量

$$O(N2^{N-1})$$ [Silander+06]

- 提案手法における計算量

$$O(N'^{k'})$$

厳密学習に比べ大幅に計算量の削減を実現

大規模な厳密学習を実現できる

数値実験

- 提案手法の有効性を検証するために，複数のベンチマークネットワークで精度評価を行った．
- 比較手法：提案手法（1/2，1.0），BDeu(ESS=1.0)[Steck+02]，CMI（閾値:0.003）[Yahezkel+09]，A*
- ベンチマークネットワーク：*bnlearn*[scutari10]に登録されている以下のネットワーク
  - cancer（変数数：5，辺数:4），Sachs（変数数：11，辺数：17），win95pts（変数数：76，辺数:112），andes(変数数:223，辺数:338)，munin（変数数：1041，辺数：1397）
- 実験手順
  1. 各ベンチマークネットワークにおいてランダムにデータを発生させる．（ネットワークによってデータ数は様々）
  2. 各手法を用いて構造学習する．
  3. 2. を各データ数において10回繰り返す．(muninのみ5回)

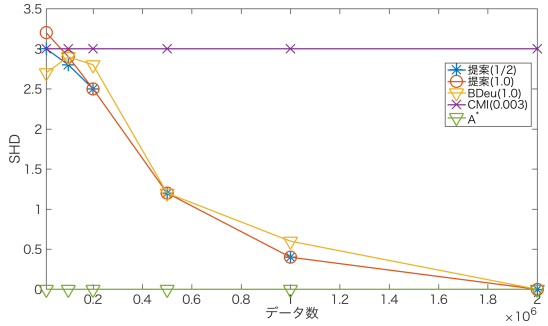
用いた評価指標

- SHD(Structural Hamming Distance)[Tsumardinos+06]:
  真の構造と学習結果の構造の距離を表す．
  - 消失辺（真では存在するが，学習によって削除した辺）
  - 余剰辺（真では存在しないが，学習によって残した辺）
  - 方向付けの誤り

によって構成

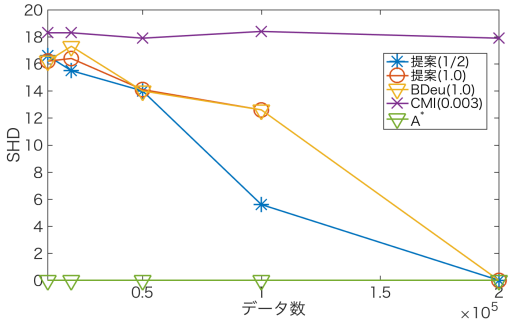
結果（cancer）

- SHD



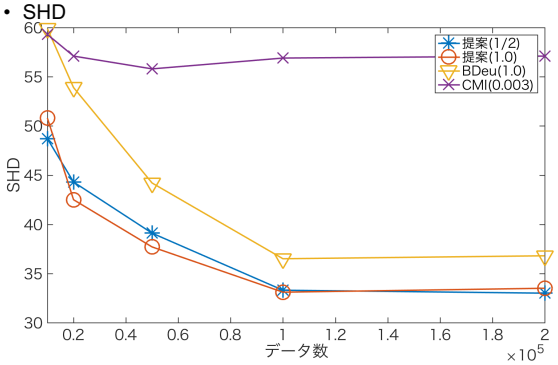
結果（sachs）

- SHD



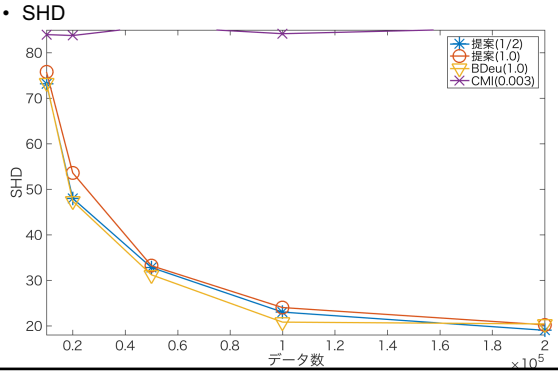
結果（win95pts）

- SHD



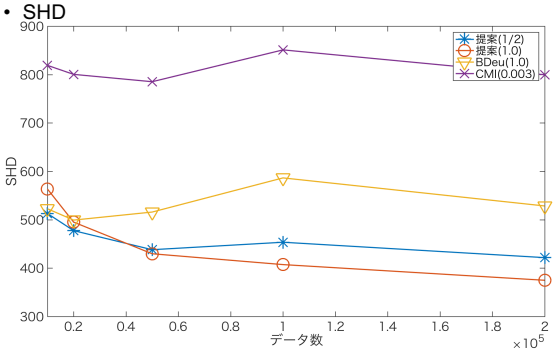
結果（andes）

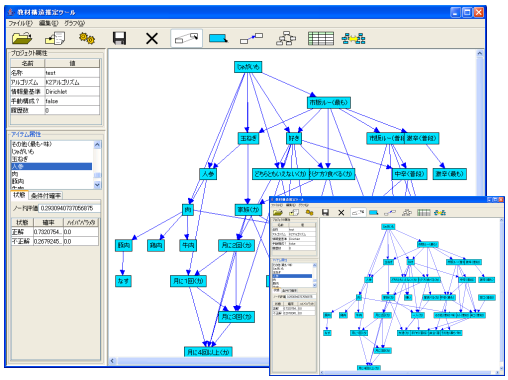
- SHD



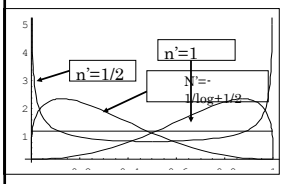
結果（munin）

- SHD



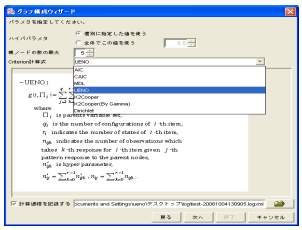


性質(1) すべてのパラメータのディレクレ分布のハイパーパラメータを設定できる



例えば、事前に因果関係があると考えられるアークの設定やないと考えられるアークの設定も事前分布の設定により矛盾なく行える

様々な情報量基準を表現できるし、それらをMIXした基準も作り出せる



本手法の特徴 （2）

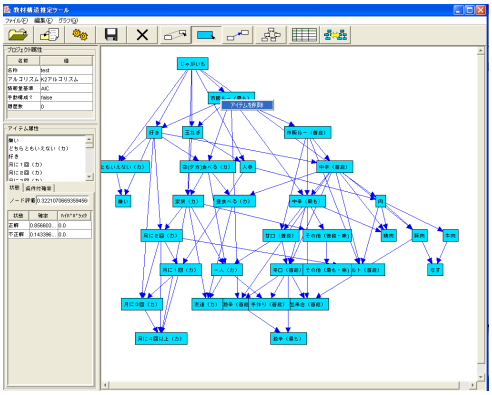
- ベイジアンネットワークモデルの各ノードの価値を定義し、評価しながら、望ましいノード集合を探索する手法

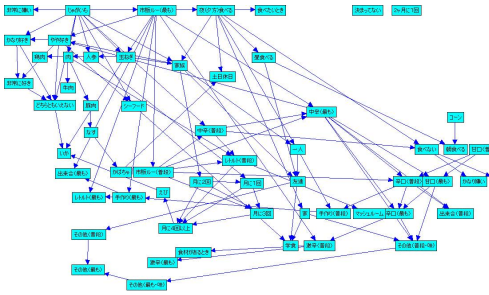
ノードの価値(Expected Value of Node Information)
植野(1993,1994,1996、1999)

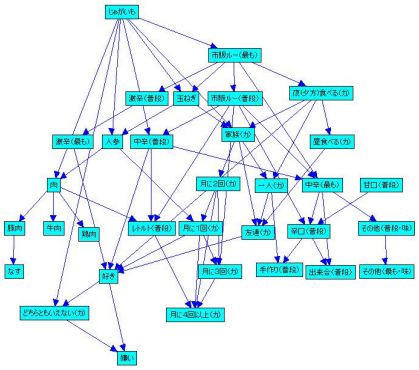
$$EVNIN = \sum_{l=1}^{\sum_{i=1}^{n} 2^{n_i}} p(x_1,\cdots x_n \mid x_i)\log p(x_1,\cdots x_n \mid x_i)$$

$$- \sum_{l=1}^{\sum_{i=1}^{n} 2^{n_i}} p(x_1,\cdots x_n)\log p(x_1,\cdots x_n)$$

変数評価機能：KeyNode







頑健で、解釈しやすくて、予測効率の高いCausal Modelが構築できる