ELSEVIER

Artificial Intelligence 137 (2002) 43–90

Artificial Intelligence

www.elsevier.com/locate/artint

Learning Bayesian networks from data: An information-theory based approach

Jie Cheng [a,*], Russell Greiner [a], Jonathan Kelly [a], David Bell [b], Weiru Liu [b]

[a] *Department of Computing Science, University of Alberta, Edmonton, AB, Canada T6G 2E8*

[b] *Faculty of Informatics, University of Ulster, UK*

Received 20 September 2000; received in revised form 13 December 2001

@mabo0725 2015年05月29日

Learning Bayesian Network from data

- 本論文はデータから大規模なベイジアン・ネットワークを構築するTPDA(Three Phase Dependency Analysis)のアルゴリズムを記述

- 2002年の発表だが、現在も大規模用BNモデルのベンチマークとして使用されている

- TPDAは「BN Power Constructor」としてフリーのソフトが公開されている。

構造推定モデルの種類

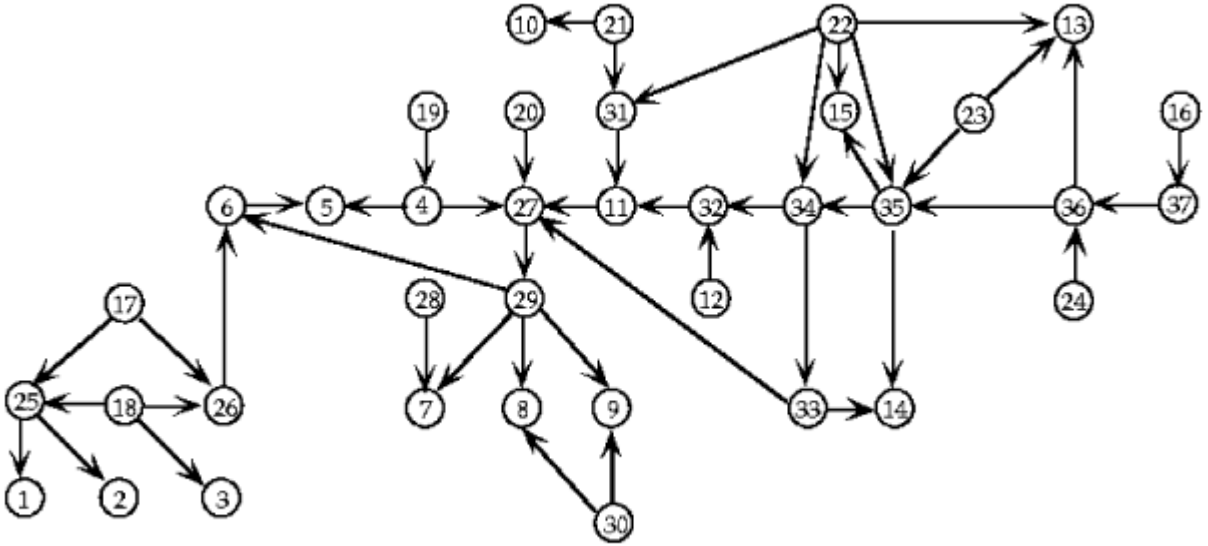


Fig. 13. The ALARM network.

- データから変数間の有意で疎な関連を図示するモデル
  - **ベイジアンネットワーク**(有向)
  - GGM(ガウシアン・グラフィカル・モデル)(無向)
  - SEM(共分散構造分析　因子分析の拡張)(有向)
  - グラフィカルLasso(無向)

BN（ベイジアンネットワーク）の課題

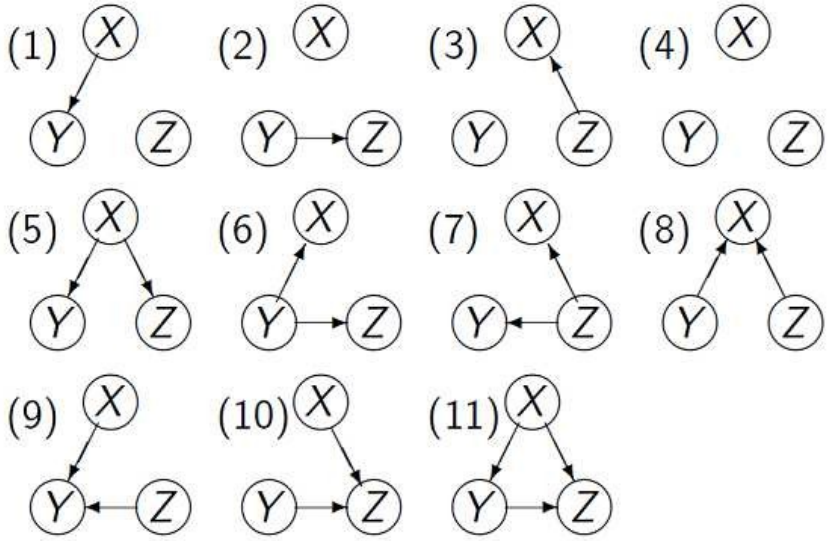
- ベイジアンネットには課題があり
  下段方向へほど問題が難しくなる。
  1. ネットワーク上での確率伝播推論
     - 確率伝播法（ノードを辿って確率が伝播する）
       風が吹けば桶屋が儲かる式の知識発見ができる
  2. データから確率分布のパラメータ推定
     - 本論ではデータの頻度で確率を計算するので言及せず
  3. データから大規模BNの構造推定（本論の課題）
  4. ベイジアンネットワークが循環する場合の対処

データからのBNの構造推定

- 現在はデータを厳密に反映した大規模BNが要求される
- 構造推定には2方法ある （現在はこの複合モデルがある）
  1. ノードのリンク状態を対数尤度指標(MDL)で計測し最適構造を求める。(score based learning)
     - MDLはAIC、BICと同様な指標
     - 組合の爆発発生し大規模BNには不適
  2. 条件付独立を検定して判定(constraint based learning)
     - 単調DAG-faithfulの概念導入（本論の前提）
       変数をノードに対応付け、条件付独立→非連結と見做す
     - TDPAは連結の増殖と縮減する**GSモデル**の1つ
     - よく利用されるPCアルゴリズムは最も簡単なモデル

MDLによる構造推定

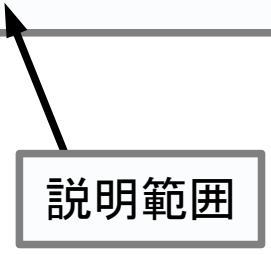
ノード間リンクの組合せの爆発



- ノード数5個　29，000組合せ
- ノード数10個　4.2×10^{18} 組合せ

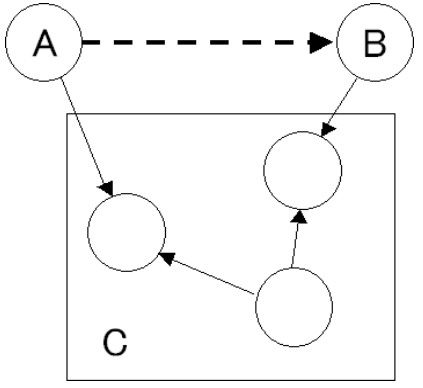
Learning Bayesian network from Dataの目次

1. 概要
2. 情報理論によるベイジアンネットの構造学習の考え方
3. SLA-Ⅱノードの順を持つ構造学習の方法の説明（TPDA-Ⅱの簡易版）
4. SLA　ノードの順が無い構造学習の方法の説明 （TPDAの簡易版)
5. TPDAとTPDA-Ⅱの説明
6. TPDAが効率的に学習していること示す実用例の紹介
7. 他のベイジアンネットワーク構造学習の方法との比較
8. TPDAの成果と将来の発展
9. 付録
   - 定理の証明
   - 単調DAG-faithful仮定の説明
   - フリーソフトの導入方法の紹介



情報理論によるBNの構造学習の考え方

- ノード間の連結は、２点間だけでなく、その間に介在するノード群 の影響も考える
- 具体的には、介在するノード群を条件とした、２点間の条件付独立を情報量で計り連結するか決定する。



I(A,B) < ε ならば XとYは独立（非連結)

$$I(A, B) = \sum_{a,b} P(a, b) \log \frac{P(a, b)}{P(a) P(b)}$$

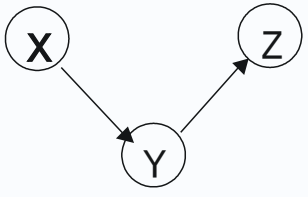
I(A,B|C) < ε ならば XとYは条件付独立 (Cを介して非連結)

$$I(A, B \mid C) = \sum_{a,b,c} P(a, b, c) \log \frac{P(a, b \mid c)}{P(a \mid c) P(b \mid c)}$$

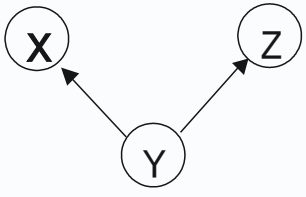
εはモデルに依存するが、0.01程度とする

条件付独立

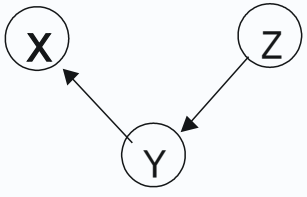
Yの条件でXとYは独立　I(X,Z|Y) ＜ε



Yが観測されると
XとYは無関係

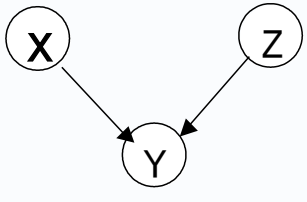


Yが観測されると
XとZは無関係



Yが観測されると
XとZは無関係

Yの条件でXとYは非独立　I(X,Z|Y)≧ε



Yが観測されると
XとZは依存関係

V構造は
矢印が決まる

SLA-II(ノード順番あり)のアルゴリズム

(1) ノードの順番で $I(X,Y) > \varepsilon$ となる組を選別し、選別リストLに入れる。
矢印の方向はX→Yとなる。

(2) 連結を増殖する過程(Thickening)
L内の(X,Y)について以下を繰返し連結を増やす。
XとYの最小の介在ノード群 C を見つける(最初はCは存在しない)
$I(X,Y|C) > \varepsilon$ ならXとYを連結する

(3) 連結を縮約する過程(Thinning)
不要な連結を条件付独立で削除する
各連結(X,Y)について以下を繰返し連結を削減する。
最小の介在ノード群 C'を見つける
$I(X,Y|C') < \varepsilon$ ならXとYを非連結とする

TDPAのアルゴリズム(1)

大規模BNに工夫されたアルゴリズム

- 最初にドラフトのBNを生成する
- 条件付独立は関数EdgeNeed_HとEdgeNeedで判断
  - EdgeNeed_Hは近傍のみで条件付独立を判断
    介在ノードの探査空間は狭いので早い
    _Hはヒュリステッィク（経験的）の意味
  - EdgeNeedは近傍の近傍で条件付独立を判断
    殆どCallされない
    介在ノードの探査空間は広く、厳密な条件独立を判断。
- ノードの矢印付けは関数OrientNodeで行う

TPDAのアルゴリズム(2)

(1) I(X,Y) > εとなるペア を選別し、I(X,Y)値の昇順にリストLにいれる。

(2) ドラフトのBNを生成（Chow-Liuの木構造BN　連結Eの生成）
　XとYを連結先はリストLから外す。

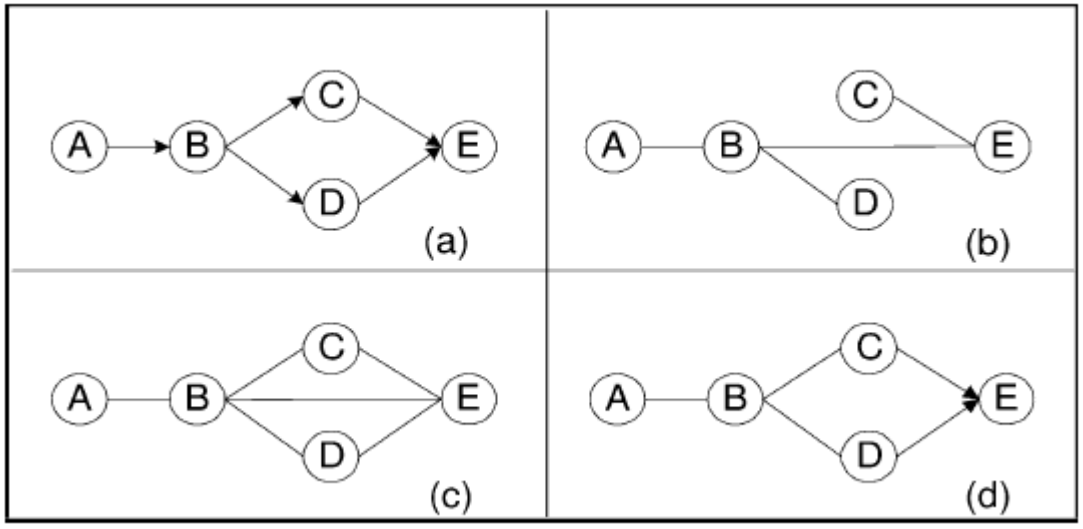
(3) 連結を増殖する過程(Thickening)
　L内(X,Y)について以下を繰返し連結を増やす。
　　関数EdgeNeed_H(X,Y,E)が 真 なら XとYを連結する。

(4) 連結を縮減する過程(Thinning)
　各連結(X,Y)について以下を繰返し連結を削減する。
　関数EdgeNeed_H(X,Y,E')が 偽 なら非連結とする

(5) まだXとYが連結している場合、以下の条件のみ行う。
　XがY以外に3近傍以上あれば or　YがX以外に3近傍あれば
　関数EdgeNeed(X,Y,E')が偽なら非連結にする

(6)関数OrientEdge(E)で方向付けする

TPDAのBN作成例



(a) 真のBN

(b) Chow-liu法によるドラフト（木構造）

(c) 連結増殖過程（B-C D-Eを連結)

(d) 連結縮約過程（B-Eを非連結)

(e) 連結の方向付け（全てに矢印は付かない）

関数OrientEdge

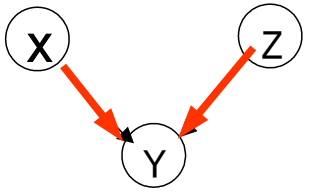
関数OrientEdge(CutSet)　３連結を見つけ方向付けを繰返す

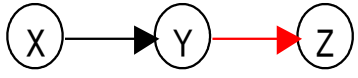
引数CutSetは関数EdgeNeed非連結にした介在ノード群
(ここでの条件独立の計算を省いている)

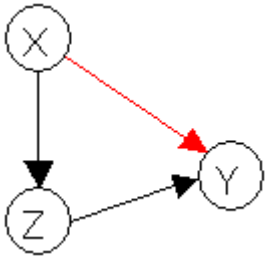
(1)V構造を見つける
　X-Yが非連結でX-Y-Zが非条件独立の場合
　CutSetに(X,Z,C')があっても(Y ∉ C')なら
　　X→Y←Z と方向付ける。
　CutSetに(X,Z,C') (Y ∈ C')が無ければ
　　X→Y←Z と方向付ける

(2)3連結(X,Y,Z)について
　X→Y－Z(X ≠ Z)ならば X→Y→Zと方向付ける

(3)X－Yの場合、XからYへ連結路があれば
　X→Yと方向付ける　(非循環にしないため)







関数NeedEdge_H(X,Y,E)

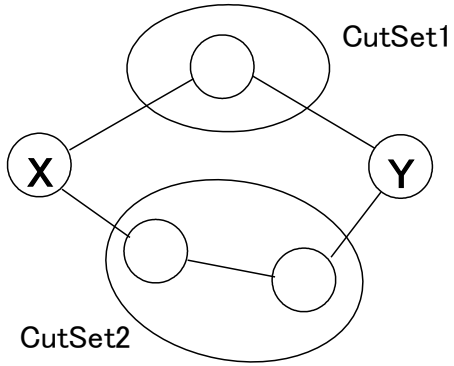
関数EdgeNeed_H(X,Y,E):連結EでのXとYの近傍で条件独立をチェック

1)Sx : XとYの連結路にありXの近傍先のノード群
  Sy : YとXの連結路にありYの近傍先のノード群
2){SxとSy}ノード群の各組合わせCについて以下を繰り返す。
  I(X,Y|C) < ε) ならば
    CutSetに(X,Y,C)を追加する (関数OrientEdgeで使う)
    偽(非連結)を返す。→ 終了

  Cのノード群について1個づつ減らして確かめる。
    C' = C群からj番目のノードを除く
    s(j) = I(X,Y|C/j)
     if(最小s(j) < ε) ならば
    CutSetリストに(X,Y,C')を追加する (関数OrientEdgeで使う)
    偽(非連結)を返す → 終了

上記以外なら2)に戻って次のノード群Cについて行う

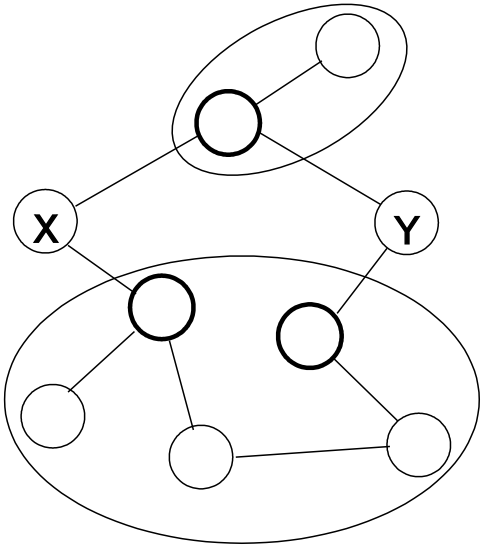
3)上記以外なら 真(連結)を返し → 終了



関数NeedEdge(X,Y,E)

関数EdgeNeed(X,Y,E):近傍の近傍まで条件独立をチェックする
- Sx : XとYの連結路にありXの近傍先のノード群
- Sy : YとXの連結路にありYの近傍先のノード群
- Sx': SxとSyの連結路にありSxノード群の近傍(Sxは含まない)
- Sy': SxとSyの連結路にありSyノード群の近傍(Syは含まない)

Sx'とSy’について関数NeedEdge_H(X,Y,E)と同じ処理をする。



まとめ(TPDA)

- TPDAはデータから大規模BNを厳密に求めるため開発されたアルゴリズム
- データの条件付独立の検定で連結を判断
- 3フェーズ(ドラフト、増殖、縮減）で構成
- ヒューリステック（経験的）な関数でBNを作成し特別な状態のみ厳密的な関数を適用する
- 連結後に関数OrientEdgeで矢印を付ける
  - J.PearlのDAG（非循環方向モデル)に準拠
    （正確な因果を表しているわでない）
  - 全ての連結に矢印が付くとは限らない

まとめ(TPDAの改良版)

- 介在ノード群の探索を3回で済む方法が考案されている(植野真臣　TPDAの高速化2010)

  ノード数1000で約1200秒((株)CAC 技術レポート)
- TPDAは大規模な範囲で条件独立を判定するので精度が劣化する

  計算時間と精度向上のため小規模に分解して構造学習するRAIの実装(森下民平 2012)がある。
- スコアベースと制約ベース(条件独立)を合体した構造学習MMHC(2006)がある。