Learning Bayesian Networks with Thousands of Variables

Mauro Scanagatta, Cassio P. De Campos, Giorgio Corani, Marco Zaffalon

Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems, 2015, December 7-12

研究のサマリー・良かった点・悪かった点

サマリー

- n(変数の個数)が大きい時にもBNの構造を推定する手法を提案
- 手法は、
  - BICが各変数に対して分解可能なことを用いて、
  - 親集合特定・構造最適化の両方で、
  - 効率的なアルゴリズムを提案

良い点

- k(in-degree)を取り払った構造の決定
- アルゴリズム効率化のための工夫(効率化しつつ、BICに基づく)

悪い点

- アルゴリズムの説明はあるが、アルゴリズム(疑似コード)がない部分がある
- アルゴリズムに焦点を置き、多変数時のモデル有用性について議論がない

新規性・有用性・信頼性

新規性

- k(in-degree)に縛られない、変数関係の推定を高速に実現 (Parent-Set Identification)
- 既存の変数順序に基づく構造最適化アルゴリズムを簡潔に改良 (Structure Optimization)

有用性

- kを定めない点で、BNの実問題への適用可能性が向上
- 多変数の構造を推定するのに足るデータ数が得られるかが怪しい

信頼性

- BICに基づきモデル・構造を評価 -> 信頼度高
- 多変数時のデータ数問題　-> 信頼度低

構成

1. Introduction (+ review)
2. Structure Learning of Bayesian Network
3. Parent set Identification
4. Structure optimization
5. Experiments
6. Conclusion and future work

提案手法

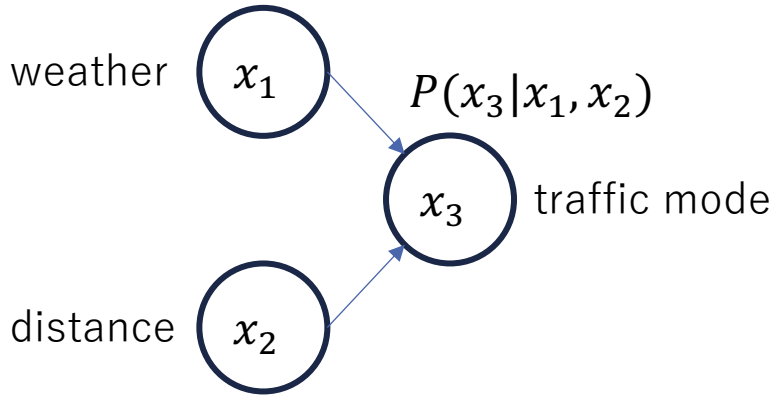
構成

1. **Introduction (+review)**
2. Structure Learning of Bayesian Network
3. Parent set Identification
4. Structure optimization
5. Experiments
6. Conclusion and future work

提案手法

補足 - ベイジアンネットワークと推定

- ベイジアンネットワーク(BN)
  - 変数間の関係をグラフとして表す機械学習モデル(グラフィカルモデル)の一種
  - Directed Acyclic Graph (DAG)のグラフを構造とする



- ネットワークの推定
  - グラフ構造(：変数の関係)
  - 周辺化確率$(P(x_3|x_1,x_2)$

  の二つを決定する必要がある。特に前者が大変。

- 構造の推定について

  全ての構造を確認し、最適なグラフ構造を選択する問題はNP困難であることが知られている。

  →本研究の対象

Introduction - 推定の方針

- Score-based learning
  「定めた指標をデータに基づき最大化する構造を探索する」ことで決定する。
  主にBayesian Information Criterion (BIC)を最大化する方針が主流。
  DPや線形計画法による厳密解法が提案されているが、遅い。

- 2 steps(主流な解き方)
  以下に示す二段階の問題を順に解くことによって、指標最大化の構造を推定する。
  1. Parent set identification
     各変数に対して、**'親変数'の集合+指標、のリスト**を生成する。
     (x_3に対して全列挙すると$[\{x_1\},\{x_2\},\{x_1,x_2\}]$の三通りで、この中から妥当なものを複数選択)
     ('親変数'：その変数を説明する変数、ある変数x_iにグラフ上でエッジが伸びる変数πの集合Π_i)

  2. Structure optimization
     各変数に対して、'親変数'の集合を決定する。=グラフを構築する。
     その際、**グラフ内に閉経路(循環)がない**ようにする。

Review

- Parent set identification

  従来手法では**k(in-degree)を設定**し、全変数組み合わせに対して指標を計算する。

  (Bartlett and Cussens, 2015), (Cussens et al., 2013)では$k = 2$を設定。

  in-degree：親集合の最大サイズ

提案手法ではこの 2 点を自由に

- Structure optimization

  ordering-based algorithm (Teyssier and Koller, 2005) が近似解法として主流。

  - 貪欲法
  - 変数間の順番を設定して、その順番内で最適な構造を決定
  - 順番を入れ替えて(貪欲法)、最適な順番&構造を探索

- K2 algorithm

  設定された変数間の順番に基づき、Parent set identification と Structure optimizationを行う

構成

1. Introduction (+review)
2. **Structure Learning of Bayesian Network**
3. Parent set Identification
4. Structure optimization
5. Experiments
6. Conclusion and future work

提案手法

Setting

- 変数設定

  N：データ数

  $\mathcal{D} = \{D_1, \ldots, D_N\}$：データ群

  n：変数数

  $\mathcal{X} = \{X_1, \ldots, X_n\}$：変数群

  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$：DAG

  $\Pi_1, \ldots, \Pi_n$：各変数の親変数集合

変数に対して分解可能
‖
変数ごとに並列化

離散化しない連続量の場合、

この部分が若干変更

- スコア関数(BIC)

$$BIC(\mathcal{G}) = \sum_{i=1}^{n} \left\{ \sum_{\pi \in \Pi_i} \sum_{x \in X_i} N_{x,\pi} \log \hat{\theta}_{x|\pi} - \frac{\log N}{2} (|X_i| - 1)(|\Pi_i|) \right\}$$

尤度の項　　ペナルティ項(モデル複雑さに対する)

$N_{x,\pi}$：$(X = x \wedge \Pi_i = \pi)$を観測した回数 <- Xが離散化されていることに注意

$\hat{\theta}_{x|\pi}$：最大尤度$P(X_i = x | \Pi_i = \pi)$

$|\cdot|$：状態の数

構成

1. Introduction (+review)
2. Structure Learning of Bayesian Network
3. **Parent set Identification**
4. Structure optimization
5. Experiments
6. Conclusion and future work

提案手法

目標

各変数に対して、**’親変数’の集合+指標、のリスト**を生成する。

X_i $\left[\left[\Pi_{i,1}, BIC_{i,1}\right], \left[\Pi_{i,2}, BIC_{i,2}\right], \ldots, \left[\Pi_{i,m}, BIC_{i,m}\right]\right]$

-> 後に、このリストの中から親変数の集合を(指標に基づき)決定する。

参照手法

- sequential ordering

  $k = 1$から始めて、$2,3, \dots, k$に対して

  1. サイズがkの親集合を全列挙
  2. 列挙した全ての親集合に対して、BICを計算しリストに加える

  (全ての組み合わせはn^kに近づく)

  $k = 1$：$[\{X_1\}, \{X_2\}, \dots \{X_n\}]$ <- ${}_nC_1$

  $k = 2$：$[\{X_1, X_2\}, \{X_1, X_3\}, \dots \{X_{n-1}, X_n\}]$ <- ${}_nC_2$

- greedy selection (提案手法の元)

  1. $k = 1$の親集合を全列挙 -> BICを計算、リストに追加
  2. 以下を時間制約まで繰り返す
     i. リストから最大BICの親集合Πを取り出す
     ii. Πに一つの変数を加えた$\{\Pi \cup X_i\}$を、 BICを計算してリストに加える(全X_iに対して)

近似指標BIC^*の導入

- 近似指標BIC^*

  BICの計算は毎回$\hat{\theta}_{x|\pi}$を含めた多くの繰り返し計算が必要 -> 計算時間を緩和したい

二つの親集合Π_1とΠ_2の和集合から成る親集合のBIC^*を、以下のように定義

$$BIC^*(X,\Pi_1,\Pi_2) = BIC(X,\Pi_1) + BIC(X,\Pi_2) + inter(X,\Pi_1,\Pi_2)$$

既知

$$\left(inter(X,\Pi_1,\Pi_2) = \frac{\log N}{2}(|X|-1)(|\Pi_1|+|\Pi_2|-|\Pi_1||\Pi_2|-1) - BIC(X,\emptyset)\right)$$

定数時間で計算可能

BIC^*の性質

- 近似精度について

Corollary 1.
Xをグラフ$\mathcal{G}$のひとつのノード、$\Pi = \Pi_1 \cup \Pi_2$をXの親集合とする。
ここで$\Pi_1 \cap \Pi_2 = \emptyset$かつ$\Pi_1$と$\Pi_2$が空集合でないとする。
このとき、

$$|BIC(X, \Pi) - BIC^*(X, \Pi_1, \Pi_2)| \leq N\ min\{H(X), H(\Pi_1), H(\Pi_2)\}$$

が成り立つ。

証明略

- 最終的に導かれる定理

Theorem 2.
Xをグラフ$\mathcal{G}$のひとつのノード、$\Pi = \Pi_1 \cup \Pi_2$をXの親集合とする。
ここで$\Pi_1 \cap \Pi_2 = \emptyset$かつ$\Pi_1$と$\Pi_2$が空集合でないとする。
このとき、ある$\Pi' \supset \Pi$に対して

$$BIC^*(X, \Pi_1, \Pi_2) + \frac{\log N}{2}(|X| - 1)|\Pi'| > N\ min\{H(X), H(\Pi_1), H(\Pi_2)\}$$

が成り立つ時、 Π'とその全ての上位集合は最適な親集合でなく、無視できる。

証明略

BIC^*の性質

- 近似精度について

> ***Corollary 1.***
> Xをグラフ$\mathcal{G}$のひとつのノード、$\Pi = \Pi_1 \cup \Pi_2$をXの親集合とする。
> ここで$\Pi_1 \cap \Pi_2 = \emptyset$かつ$\Pi_1$と$\Pi_2$が空集合でないとする。
> このとき、
> $$|BIC(X,\Pi) - BIC^*(X,\Pi_1,\Pi_2)| \leq N\ min\{H(X), H(\Pi_1), H(\Pi_2)\}$$
> が成り立つ。

- 利用方法

$BIC(X,\Pi_1)$や$BIC(X,\Pi_2)$を計算する際に、 $H(\Pi_1)$と$H(\Pi_2)$を同ループ内で計算することができる。

これにより、 $BIC(X,\Pi)$から大きなズレがないことを確認しながら、 $BIC^*(X,\Pi_1,\Pi_2)$を使える。

BIC^*の性質

- 最終的に導かれる定理

> ***Theorem 2.***
> Xをグラフ$\mathcal{G}$のひとつのノード、$\Pi = \Pi_1 \cup \Pi_2$をXの親集合とする。
> ここで$\Pi_1 \cap \Pi_2 = \emptyset$かつ$\Pi_1$と$\Pi_2$が空集合でないとする。
> このとき、ある$\Pi' \supset \Pi$に対して
> $$BIC^*(X, \Pi_1, \Pi_2) + \frac{\log N}{2}(|X| - 1)|\Pi'| > N\ min\{H(X), H(\Pi_1), H(\Pi_2)\}$$
> が成り立つ時、 Π'とその全ての上位集合は最適な親集合でなく、無視できる。

- 利用方法

BIC^*を計算済みの親集合Πに、変数を一つ加えてできたΠ'に対して***Theorem 2.***を確認することで、
Π'と、 Π'に変数を加えていって作れる親集合を考慮しなくて良い(枝切りできる)かを判定する。

Independence selection algorithm

- 用意するリスト

  $open$：親集合, BIC^*を格納する。探索対象の親集合をBIC^*順に格納。

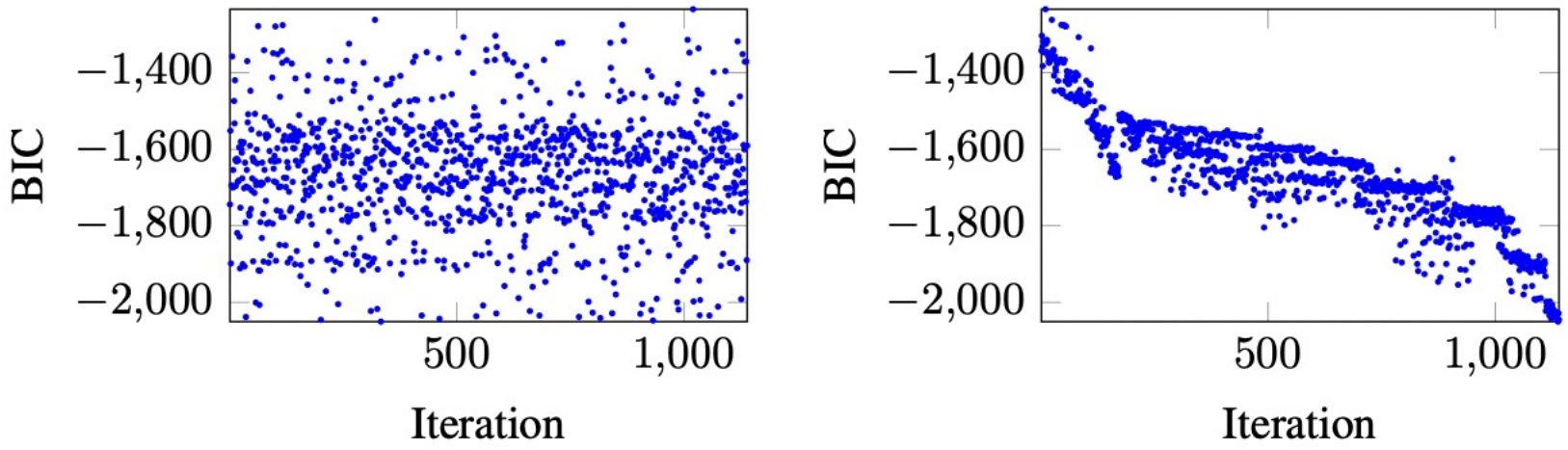
  $closed$：親集合,BICを格納する。

- アルゴリズム

  1. $k=1$の親集合を全列挙 -> BICを計算、 $closed$に追加
  2. $k=2$の親集合を、$closed$を参照しながらBIC^*を計算、 $open$に追加
  3. 以下を繰り返す($while\ open \neq \emptyset$)か、時間制約まで)
     - i. 最大BIC^*の親集合Πを$open$から取り出す
     - ii. ΠのBICを計算し、 $closed$に追加
     - iii. ひとつの変数Yを加えた未探索親集合$\Pi' = \Pi \cup Y$に対して、 BIC^*を計算
     - iv. Π'に対して ***Theorem 2.***を確認し、満たさなければ$open$に追加
  4. $closed$を出力

BIC^*が大きい親集合へ探索を広げることが可能

提案アルゴリズムの挙動



(a) Sequential ordering.

(b) Indep. selection ordering.

Figure 1: Exploration of the parent sets space for a given variable performed by sequential ordering and independence selection. Each point refers to a distinct parent set.

1 iterationにかかる計算量が違う、1 iteration何秒かという設定が何もないので怪しいが、探索を進める様子は確認できる。

構成

1. Introduction (+review)
2. Structure Learning of Bayesian Network
3. Parent set Identification
4. **Structure optimization**
5. Experiments
6. Conclusion and future work

提案手法

目標

生成した**’親変数’の集合+指標、のリスト**を用いて、グラフを構築する。

$$\left[[\Pi_{i,1}, BIC_{i,1}], [\Pi_{i,2}, BIC_{i,2}], \ldots, [\Pi_{i,m}, BIC_{i,m}]\right] \qquad \mathcal{G}$$

＝各変数に対して、親変数の集合を決定する。
グラフは取りうるDAGの中で、最高スコアのものを目指す。

参照手法

- Ordering based search (OBS) (Teyssien & Koller, 2005)
  ① ある変数間順序に従った最適グラフを構築 (貪欲法) -> K2アルゴリズム (Cooper & Herskovits, 1992)
  ② 最適グラフを返す変数間順序を決定 (貪欲法) -> greedy hill algorithm
  という、 2 階層の探索アルゴリズム。

  ‘変数間順序’の軸方向に探索する点が画期的

- 変数間順序
  変数間に、親関係の順序を定める
  $(X_1, X_2, \dots, X_n)$という順序なら、 X_3の親集合は$\{X_1\}, \{X_2\}, \{X_1, X_2\}$のどれか
  -> 絶対にacyclicなグラフを作成できる

- 変数間順序に関する貪欲法
  $(X_1, X_2, \dots, X_j, X_{j+1} \dots, X_n) \mapsto (X_1, X_2, \dots, X_{j+1}, X_j \dots, X_n)$の (n-1)通りあるswapを行い、
  最大のswap結果に更新していく。

Acyclic selection OBS

- input

  parent_sets：parent set identificationで作成した、[親集合, BIC]のリスト。

$(X_1, X_2, \ldots, X_n)$という変数の順番であると考える。

1. $N \times N$のbool型行列mを、全てFalseで初期化。($m(\mathrm{X}, \mathrm{Y})$は$Y$が$X$の**子孫**変数であるかを格納)
2. for 全変数$X_i, (i = \mathrm{n}, \mathrm{n}-1, \ldots, 1)$　　⊃ 子変数
   - i. parent_sets内で、 V_iの子変数を含まない&最高BICの親集合を探索する。
   - ii. $X_j (j = 1, \ldots, i)$に空のリスト'todo'を割り当てる。
   - iii. X_iの'todo'リストに、 $m(\mathrm{X}, \mathrm{Y})$を参照して$X_i$の子孫変数を加える。
   - iv. 全てのX_iの**先祖**変数XをDFSで探索し、以下を実行。
     - a. Xの'todo'リスト内の全ての変数Yに対して、

       $m(\mathrm{X}, \mathrm{Y})$がtrue -> continue。

       $m(\mathrm{X}, \mathrm{Y})$がfalse -> $m(\mathrm{X}, \mathrm{Y})$をtrueに設定し、Yを$X$の親集合の'todo'に追加する。

OBSとの比較

- 順序$(X_1, X_2, \dots, X_n)$を考慮する際に出力される構造
  OBS：順序に従った構造(X_2はX_1の親,先祖変数にはなり得ない)
  ASOBS：順番に(逆から)定められていった構造
  -> X_2がX_1の親,先祖変数になるといったback-arcを考慮しうる
  -> ASOBSでは、OBSで考慮している(限られた)構造を、包括した構造を考慮できる

- 計算量
  OBS：$O(Ck)$
  ASOBS：$O(Ck + n^2)$　　$C = \sum_{i=1}^{n} c_i \, , where \; c_i \; is \; len(\text{parent\_sets}), \& \; usually \; C > n^2/k$

1. $N \times N$のbool型行列mを、全てFalseで初期化。($m(\mathrm{X}, \mathrm{Y})$は$Y$が$X$の**子孫**変数であるかを格納)
2. for 全変数$X_j, (\mathrm{j} = \mathrm{n}, \mathrm{n}-1, \dots, 1)$
   i. parent_set内で、 V_jの子変数を含まない&最高BICの親集合を探索する。
   ii. $X_i (i = 1, \dots, j)$に空のリスト'todo'を割り当てる。
   iii. X_jの'todo'リストに、 $m(\mathrm{X}, \mathrm{Y})$を参照して$X_j$の子孫変数を加える。
   iv. 全てのX_jの**先祖**変数XをDFSで探索し、以下を実行。
      a. Xの'todo'リスト内の全ての変数Yに対して、
         $m(\mathrm{X}, \mathrm{Y})$がtrue -> continue。
         $m(\mathrm{X}, \mathrm{Y})$がfalse -> $m(\mathrm{X}, \mathrm{Y})$をtrueに設定し、Yを$X$の親集合の'todo'に追加する。

出力に関する主張

Theorem 3.

ASOBSがある順序$(X_1, X_2, \ldots, X_n)$が与えられた時に出力するグラフ$\mathcal{G}$のスコアは、OBSが出力するグラフのスコアと同等かそれ以上。

-> 順序が与えられた時にOBSと同等以上

Theorem 4.

OBSが順序のswap$(X_1, X_2, \ldots, X_j, X_{j+1} \ldots, X_n) \longmapsto (X_1, X_2, \ldots, X_{j+1}, X_j \ldots, X_n)$によってスコアを改善する時、ASOBSも同じswapを採用し、スコアを改善することが可能である。

-> 順序の探索をOBSと同等に行える

構成

1. Introduction (+review)
2. Structure Learning of Bayesian Network
3. Parent set Identification
4. Structure optimization

提案手法

5. **Experiments**
6. Conclusion and future work

Settings

- 検証手法

  parent set identification

  　　：**independence selection(IS)**, sequential selection(SQ), greedy selection(GS)の3つ

  structure optimization

  　　：**ASOBS**, OBS, Gobnilp(exact solver)の3つ

  これら二段階の組み合わせ($3\times3=9$通り)を比較する。

- 比較手法

  parent set identificationには変数毎$1min$、structure optimizationには合計$24hour$の制約内で

  　　見つけられた最高BICのグラフを出力する。

  2つの手法で構築したグラフのBICの差を計算し、

  $$\{0\sim2, 2\sim6, 6\sim10, over\ 10\} \rightarrow \{neutral, positive, strong, very\ strong\}$$

  と分類する。

  これを、複数のデータセットに対して行い、$\{neutral, positive, strong, very\ strong\}$にそれぞれいくつ割り当てられるかを確認する。

- open dataのケース

Data set	n	Data set	n	Data set	n	Data set	n
Audio	100	Retail	135	MSWeb	294	Reuters-52	889
Jester	100	Pumsb-star	163	Book	500	C20NG	910
Netflix	100	DNA	180	EachMovie	500	BBC	1058
Accidents	111	Kosarek	190	WebKB	839	Ad	1556

Table 1: Data sets sorted according to the number n of variables.

これらをランダムに3分割づつし、48データセットで評価

※分割する際にはデータサイズnを大きく保つために、重なりを許容

結果①

- parent set identificationの比較結果
  - IS(提案手法)が、他手法より強い
  - 特に厳密解法であるGobnilpと組み合わせることで、
    提案手法の強みである、高BICを持つ親集合の探索が効果を発揮している

structure solver	Gobnilp		ASOBS		OBS	
parent identification: IS vs	GS	SQ	GS	SQ	GS	SQ
ΔBIC (K)						
Very positive (K >10)	44	38	44	30	44	32
Strongly positive (6<K <10)	0	0	0	4	1	0
Positive (2 <K <6)	0	4	2	3	0	2
Neutral (-2 <K <2)	2	3	0	4	2	4
Negative (-6 <K <-2)	0	1	2	1	0	2
Strongly negative (-10 <K <-6)	1	1	0	5	0	4
Very negative (K <-10)	1	1	0	1	1	4
p-value	**<0.01**	**<0.01**	**<0.01**	**<0.01**	**<0.01**	**<0.01**

Table 2: Comparison of the approaches for parent set identification on 48 data sets. Given any fixed solver for structural optimization, IS results in significantly higher BIC scores than both GS and SQ.

結果②

- structure optimizationの比較結果
  - 下の表は大きいサイズ($n > 500$)のデータセットのみでの比較であることに注意
  - $n = 10$程度のデータでは、厳密解法のGobnilpが当然最高スコアを出した

parent identification	**Independence sel.**		**Forward sel**		**Sequential sel.**	
structure solver: AS vs	GP	OB	GP	OB	GP	OB
ΔBIC (K)						
Very positive (K >10)	21	21	20	21	19	21
Strongly positive (6<K<10)	0	0	0	0	0	0
Positive (2<K<6)	0	0	0	0	0	0
Neutral (-2<K<2)	0	0	0	0	0	0
Negative (-6<K<-2)	0	0	0	0	0	0
Strongly negative (-10<K<-6)	0	0	0	0	0	0
Very negative (K<-10)	0	0	1	0	2	0
p-value	**<0.01**	**<0.01**	**<0.01**	**<0.01**	**<0.01**	**<0.01**

Table 3: Comparison between the structure optimization approaches on the 21 data sets with $n \geq 500$. ASOBS (AS) outperforms both Gobnilp (GB) and OBS (OB), under any chosen approach for parent set identification.

自作データでの実験

- $n = 2000, 4000, 10000$の自作モデル5つずつ(計15つ)+$n = 100$~1000程度の公開モデル
  からデータを生成し($N = 5000$)、同様に手法ごとのBIC比較を行なった

- 結果、parent set identification, structure optimization共に提案手法が
  比較手法を大きく上回った

構成

1. Introduction (+review)
2. Structure Learning of Bayesian Network
3. Parent set Identification
4. Structure optimization

提案手法

5. Experiments
6. **Conclusion and future work**

まとめ

- 目的
  大きなサイズ$(n \geq 1000)$のベイジアンネットワークを構築するアルゴリズムを開発

- 手法
  Parent set identification：k(in-degree)を取り除いて最適な親集合を効率的に探索するアルゴリズム
  Structure optimization　：変数が多い時に用いる、変数の順番に従った探索方法を拡張し、制限緩和

- 結果
  多変数のデータセットで、両アルゴリズムによって得られるグラフが高BICスコアであることを確認
  (時間制限内で良い構造を見つけられる、という比較)

研究への適応

- activity-simの構造決定をデータ駆動にするため、

構造化されたBN(又は類するGM)を考えている

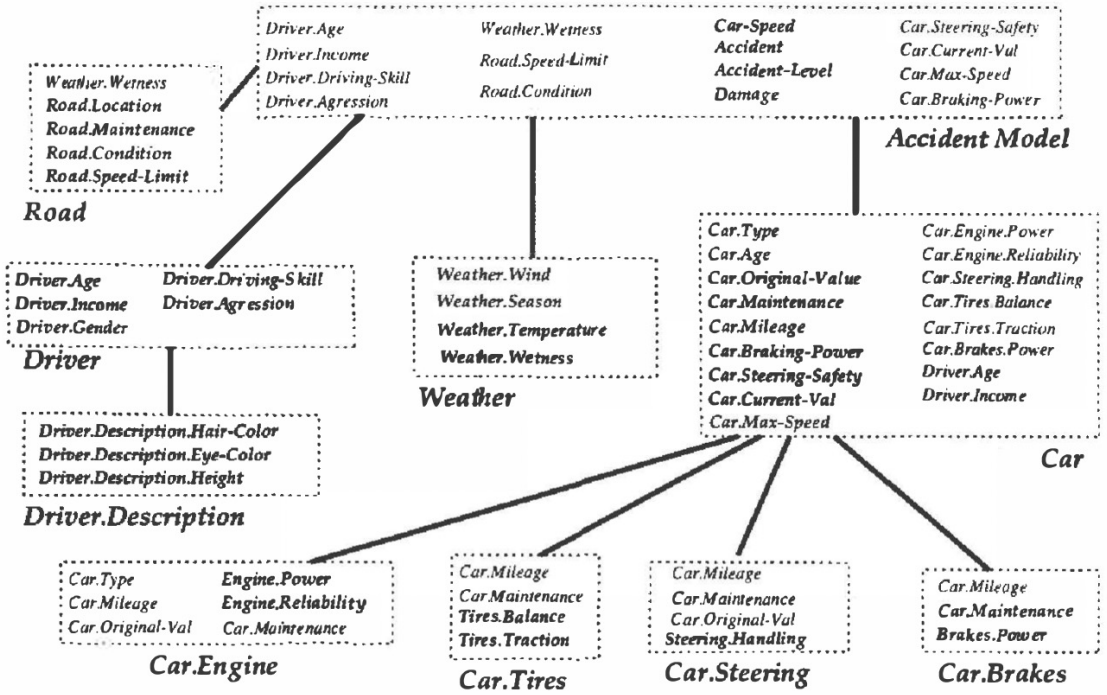
  - objectは、人・ツアー(とその特徴)・場所など

- 構造を決定するために、
  - スコアを近似
  - 探索方向を工夫する(今回は貪欲法、枝切り)

といったことは、機械学習と相性が良い

- どんなグラフィカルモデルを組むにしても、

構造を決定する手法の枠組み、簡略化は参考になる



Koller & Pfeffer, 1997