FAST BAYESIAN NETWORK STRUCTURE LEARNING WITH QUASI-DETERMINISM SCREENING

Journées Francophones sur les Réseaux Bayesiens, INRA Toulouse

Thibaud Rahier, Sylvain Marié, Stéphane Girard, Florence Forbes
May 31, 2018

INRIA - Schneider Electric
I - Bayesian network structure learning

II - Determinism and Bayesian networks

III - Structure learning with (quasi-)determinism screening

IV - Experiments

V - Discussion
I - Bayesian network structure learning

II - Determinism and Bayesian networks

III - Structure learning with (quasi-)determinism screening

IV - Experiments

V - Discussion
I-1. BAYESIAN NETWORKS

Setting

- \((X_1, \ldots, X_n)\): tuple of categorical random variables
- \(D\): dataset containing \(M\) i.i.d instances of \((X_1, \ldots, X_n)\)
I-1. BAYESIAN NETWORKS

Setting

- \((X_1, \ldots, X_n)\): tuple of categorical random variables
- \(D\): dataset containing \(M\) i.i.d instances of \((X_1, \ldots, X_n)\)

Bayesian network: \(B = (G, \theta)\) where

- \(G = (V, A)\): DAG structure with
  - \(V = \{1, \ldots, n\}\) vertices associated to the \(n\) variables
  - \(A \subset V^2\) set of arcs
  - \(\pi_i\) the set of parents of \(i\) in \(G\)

Factorization of the joint distribution:

\[
P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | X_{\pi_i})
\]

- \(\theta\): parameters of the local \(P(X_i | X_{\pi_i})\)
Score&search-based BN structure learning

For a scoring function \( s : \text{DAG}_V \rightarrow \mathbb{R} \), BNSL\(_s\) comes down to:

\[
\hat{G} \in \text{argmax} \ s(G) \\
G \in \text{DAG}_V
\]
Score&search-based BN structure learning

For a scoring function $s : \text{DAG}_V \rightarrow \mathbb{R}$, BNSL$_s$ comes down to:

$$\hat{G} \in \operatorname{argmax} s(G) \quad \text{subject to} \quad G \in \text{DAG}_V$$

Some scoring functions

Most scoring functions are based on the log-likelihood $l(\theta : D)$:

$$l(\theta : D) = \sum_{m=1}^{M} \sum_{i=1}^{n} \log \left( \theta_{x_i[m] | x_{\pi_i[m]}} \right)$$

As the MaxLogLikelihood score (MLL), (leads to complete graphs):

$$s^{\text{MLL}}(G : D) = \max_{\theta \in \Theta_G} l(\theta : D)$$

In practice, we rather use regularized scores such as BIC, AIC or BDe
I - Bayesian network structure learning

II - Determinism and Bayesian networks

III - Structure learning with (quasi-)determinism screening

IV - Experiments

V - Discussion
Conditional Shannon entropy

The conditional Shannon entropy of $X_i$ knowing $X_j$ is defined as

$$H(X_i|X_j) = - \sum_{x_i, x_j} p(x_i, x_j) \log(p(x_i|x_j))$$

$H(X_i|X_j) = 0$ if and only if the value of $X_i$ is entirely determined by the value of $X_j$
Conditional Shannon entropy

The conditional Shannon entropy of $X_i$ knowing $X_j$ is defined as

$$H(X_i | X_j) = - \sum_{x_i, x_j} p(x_i, x_j) \log(p(x_i | x_j))$$

$H(X_i | X_j) = 0$ if and only if the value of $X_i$ is entirely determined by the value of $X_j$

Linking the entropy with MLL score

The MLL score can be rewritten as

$$s^{\text{MLL}}(G : D) = -M \sum_{i=1}^{n} H^D(X_i | X_{\pi_i})$$
II-2. DETERMINISM

Definitions: determinism and quasi-determinism

The relationship $X_i \rightarrow X_j$ is **deterministic** wrt $D$ iff

$$H^D(X_i|X_j) = 0$$

The relationship $X_i \rightarrow X_j$ is $\epsilon$–**quasi deterministic** wrt $D$ iff

$$H^D(X_i|X_j) \leq \epsilon$$

Definition: deterministic graphs

A **DAG** $G$ is deterministic wrt $D$ iff for every $i \in V$ st $\pi_i \neq \emptyset$,

$$H^D(X_i|X_{\pi_i}) = 0$$

(analogous definition for quasi-deterministic DAGs)
Proposition 1: Deterministic trees and the MLL score

If $T \in \text{DAG}_V$ is a deterministic tree (single-parented DAG) wrt $D$ then $T$ is a solution of $\text{BNSL}_{\text{MLL}}$:

$$s_{\text{MLL}}(T : D) = \max_{G \in \text{DAG}_V} s_{\text{MLL}}(G : D)$$
Proposition 1: Deterministic trees and the MLL score

If $T \in \text{DAG}_V$ is a deterministic tree (single-parented DAG) wrt $D$ then $T$ is a solution of $\text{BNSL}_{\text{MLL}}$:

$$s^{\text{MLL}}(T : D) = \max_{G \in \text{DAG}_V} s^{\text{MLL}}(G : D)$$

Proposition 2: Deterministic forests and the MLL score

Let $F \in \text{DAG}_V$ be a deterministic forest, and $R(F) \subset V$ its roots. If $G_R$ is a solution of $\text{BNSL}_{\text{MLL}}$ on $\{X_j, j \in R(F)\}$, then $F \cup G_R$ is a solution of $\text{BNSL}_{\text{MLL}}$ on $\{X_1, \ldots, X_n\}$:

$$s^{\text{MLL}}(F \cup G_R : D) = \max_{G \in \text{DAG}_V} s^{\text{MLL}}(G : D)$$
I - Bayesian network structure learning

II - Determinism and Bayesian networks

III - Structure learning with (quasi-)determinism screening

IV - Experiments

V - Discussion
Summary of the theoretical results

- If we can relate all variables by a single deterministic tree, then this tree is an optimal solution to $\text{BNSL}_\text{MLL}$.
- If we can relate subsets of the variables by deterministic trees, solving $\text{BNSL}_\text{MLL}$ narrows down to the roots of the trees.

→ Let’s search for deterministic subtrees before solving BNSL!
Summary of the theoretical results

• If we can relate all variables by a single deterministic tree, then this tree is a optimal solution to $\text{BNSL}_{\text{MLL}}$

• If we can relate subsets of the variables by deterministic trees, solving $\text{BNSL}_{\text{MLL}}$ narrows down to the roots of the trees

→ Let’s search for deterministic subtrees before solving BNSL!

What if the target BNSL score is not MLL score?

**Intuition:** trees have very small complexity and are therefore also interesting wrt scores such as $\text{BIC}$ or $\text{BDe}$. 
III-1. (QUASI-)DETERMINISTIC SCREENING: IDEA

Summary of the theoretical results

- If we can relate all variables by a single deterministic tree, then this tree is an optimal solution to $BNSL_{\text{MLL}}$.
- If we can relate subsets of the variables by deterministic trees, solving $BNSL_{\text{MLL}}$ narrows down to the roots of the trees.

→ Let’s search for deterministic subtrees before solving BNSL!

What if the target BNSL score is not MLL score?

Intuition: trees have very small complexity and are therefore also interesting wrt scores such as BIC or BDe.

What about quasi-determinism?

Empirical determinism is rare, however very strong relationships (i.e. very low conditional entropies) are common.

→ Let’s search for quasi-deterministic subtrees before solving BNSL!
Algorithm 1 Bayesian network structure learning with quasi deterministic screening (qds-BNSL)

**Input:** D, \( \epsilon \), sota-BNSL

1. Compute \( F_\epsilon \) by running **qd-screening** with D and \( \epsilon \)
2. Identify \( R(F_\epsilon) = \{i \in [1, n] \mid \pi^{F_\epsilon}(i) = \emptyset\} \), the set of \( F_\epsilon \)'s roots.
3. Compute \( G^*_{R(F_\epsilon)} \) by running sota-BNSL on \( X_{R(F_\epsilon)} \)
4. \( G_\epsilon^* \leftarrow F_\epsilon \cup G^*_{R(F_\epsilon)} \)

**Output:** \( G_\epsilon^* \)
Algorithm 2 Bayesian network structure learning with quasi deterministic screening (qds-BNSL)

**Input:** \( D, \epsilon, \text{sota-BNSL} \)

1. Compute \( F_\epsilon \) by running **qd-screening** with \( D \) and \( \epsilon \)
2. Identify \( R(F_\epsilon) = \{ i \in [1, n] \mid \pi^{F_\epsilon}(i) = \emptyset \} \), the set of \( F_\epsilon \)'s roots.
3. Compute \( G^*_R(F_\epsilon) \) by running sota-BNSL on \( X_{R(F_\epsilon)} \)
4. \( G^*_\epsilon \leftarrow F_\epsilon \cup G^*_R(F_\epsilon) \)

**Output:** \( G^*_\epsilon \)
Algorithm 3 Bayesian network structure learning with quasi deterministic screening (qds-BNSL)

**Input:** $D$, $\epsilon$, sota-BNSL

1. Compute $F_\epsilon$ by running **qd-screening** with $D$ and $\epsilon$
2. Identify $R(F_\epsilon) = \{i \in [1, n] | \pi^{F_\epsilon}(i) = \emptyset\}$, the set of $F_\epsilon$’s roots.
3. Compute $G^*_R(F_\epsilon)$ by running sota-BNSL on $X_{R(F_\epsilon)}$
4. $G^*_\epsilon \leftarrow F_\epsilon \cup G^*_R(F_\epsilon)$

**Output:** $G^*_\epsilon$

**Complexity**

- **qd-screening:** $O(n^2)$
- **qds-BNSL:** calls sota-BNSL on $|R(F_\epsilon)| \leq n$ variables (exact BNSL: $O(2^p)$, heuristics are very time-intensive as well)

We expect qds-BNSL to be faster than sota-BNSL when $R(F_\epsilon) < n$
I - Bayesian network structure learning

II - Determinism and Bayesian networks

III - Structure learning with (quasi-)determinism screening

IV - Experiments

V - Discussion
IV-1. BAYESIAN NETWORKS LEARNT ON THE MSNBC DATASET: BASELINE

BN learnt on dataset 'msnbc' with sota-BNSL
IV-1. BAYESIAN NETWORKS LEARNT ON THE MSNBC DATASET: QDS

BN learnt on dataset 'msnbc' with qds–BNSL (eps_0.5)
CVLogLikelihood score VS NbArcs for different sparsity induction methods

-6.30  -6.25  -6.20  -6.15  -6.10  -6.05
70  80  90  100

Better generalization perf.

Worse generalization perf.

Better readability

Worse readability

SparsityInductionMethod
- EquivalentSampleSizeDecreasing
- NbParentsRestriction
- QuasiDeterminismScreening
Computation Time VS CVLL score for different sparsity induction methods

- EquivalentSampleSizeDecreasing
- NbParentsRestriction
- QuasiDeterminismScreening
IV-4. PERFORMANCE/READABILITY TRADEOFF - PIU DATASET

CVLogLikelihood score VS NbArcs for different sparsity induction methods

- better generalization perf.
- worse generalization perf.

SparsityInductionMethod
- EquivalentSampleSizeDecreasing
- NbParentsRestriction
- QuasiDeterminismScreening

Better readability
Worse readability
IV-5. PERFORMANCE/COMPUTATION TIME TRADEOFF - PIU DATASET

Computation Time VS CVLL score for different sparsity induction methods

SparsityInductionMethod
- EquivalentSampleSizeDecreasing
- NbParentsRestriction
- QuasiDeterminismScreening

faster
slower
better generalization perf.
worse generalization perf.
I - Bayesian network structure learning

II - Determinism and Bayesian networks

III - Structure learning with (quasi-)determinism screening

IV - Experiments

V - Discussion
Summary

- Deterministic screening is consistent wrt the MLL score.
- BN learnt via qds-BNSL have often have a very interesting performance-vs-readability tradeoff, and are consistently faster to compute for a given performance score than with usual methods.

However these properties depend highly on the dataset.
Summary

- Deterministic screening is consistent wrt the MLL score
- BN learnt via qds-BNSL have often have a very interesting performance-vs-readability tradeoff, and are consistently faster to compute for a given performance score than with usual methods

However these properties depend highly on the dataset

Perspectives

In the future we plan to

- Search for guarantees of qds-BNSL wrt scores as BIC, BDe or CVLL
- Look for a criteria that enables us to choose $\epsilon$ in a principled way
Comparison of the proportion of variables remaining after screening to the repartition of entropies

Legend

- RemainingVariables
- EntropiesRepartion
Comparison of the proportion of variables remaining after screening to the repartition of entropies.
Thank you
More results
CVLogLikelihood score VS NbArcs for different sparsity induction methods

-6.10 -6.05 -6.15 -6.20 -6.25 -6.30 -6.35

VLLScore

70 80 90 ... induction methods

better generalization perf.

worse generalization perf.

worse readability better readability

SparsityInductionMethod

EquivalentSampleSizeDecreasing

NbParentsRestriction

QuasiDeterminismScreening

better generalization perf.

worse readability
APP 1. PERFORMANCE/READABILITY TRADEOFF - MSNBC DATASET (2/2)

CVLogLikelihood score VS NbArcs for different sparsity induction methods

-6.6
-6.4
-6.2
25 50 75 ... induction methods
better generalization perf.

worse generalization perf.

better readability
worse readability

worse generalization perf.

SparsityInductionMethod
- EquivalentSampleSizeDecreasing
- NbParentsRestriction
- QuasiDeterminismScreening

better generalization
worse generalization
better readability
worse readability

perf.
Computation Time VS CVLL score for different sparsity induction methods

- Faster generalization perf.
- Slower generalization perf.

SparsityInductionMethod:
- EquivalentSampleSizeDecreasing
- NbParentsRestriction
- QuasiDeterminismScreening

VLLScore
APP 3. PERFORMANCE/READABILITY TRADEOFF - PIU DATASET

CVLogLikelihood score VS NbArcs for different sparsity induction methods

- better generalization perf.
- worse generalization perf.
- better readability
- worse readability

SparsityInductionMethod:
- EquivalentSampleSizeDecreasing
- NbParentsRestriction
- QuasiDeterminismScreening

NbArcs

VLLScore
Computation Time VS CVLL score for different sparsity induction methods

- EquivalentSampleSizeDecreasing
- NbParentsRestriction
- QuasiDeterminismScreening

The graph shows the tradeoff between computation time and CVLL score for different sparsity induction methods. The x-axis represents the VLL Score, and the y-axis represents the computation time. The graph indicates that faster methods generally result in better generalization performance, while slower methods lead to worse generalization performance.
Algorithm 4 Quasi-determinism screening (qds)

Input: \( D, \epsilon \)

1. Compute empirical cond. entropy matrix \( H^D = (H^D(X_i|X_j))_{1 \leq i,j \leq n} \)
2. for \( i = 1 \) to \( n \) do
3. compute \( \pi_\epsilon(i) = \{ j \in [1, n] \setminus \{i\} \mid H^D_{ij} \leq \epsilon \} \)
4. for \( i = 1 \) to \( n \) do
5. if \( \exists j \in \pi_\epsilon(i) \) s.t. \( i \in \pi_\epsilon(j) \) then
6. if \( H^D_{ij} \leq H^D_{ji} \) then \( \pi_\epsilon(j) \leftarrow \pi_\epsilon(j) \setminus \{i\} \)
7. else \( \pi_\epsilon(i) \leftarrow \pi_\epsilon(i) \setminus \{j\} \)
8. for \( i = 1 \) to \( n \) do
9. \( \pi^*_\epsilon(i) \leftarrow \arg\min_{j \in \pi_\epsilon(i)} |Val(X_j)| \)
10. Compute forest \( F_\epsilon = (V_{F_\epsilon}, A_{F_\epsilon}) \), where
    - \( V_{F_\epsilon} = [1, n] \)
    - \( A_{F_\epsilon} = \{((\pi^*_\epsilon(i), i) \mid i \in [1, n] \text{ s.t. } \pi^*_\epsilon(i) \neq \emptyset} \)

Output: \( F_\epsilon \)
CVLogLikelihood score VS NbArcs for different sparsity induction methods

better generalization perf.

worse generalization perf.

worse readability

better readability

SparsityInductionMethod

- EquivalentSampleSizeDecreasing
- NbParentsRestriction
- QuasiDeterminismScreening

better readability
worse readability
worse generalization perf.
better generalization perf.
Computation Time VS CVLL score for different sparsity induction methods

SparsityInductionMethod
- EquivalentSampleSizeDecreasing
- NbParentsRestriction
- QuasiDeterminismScreening

faster
slower
better generalization perf.
worse generalization perf.
Comparison of the proportion of variables remaining after screening to the repartition of entropies

Legend
- RemainingVariables
- EntropiesRepartition