



HAL
open science

Screening strong pairwise relationships for fast Bayesian network structure learning

Thibaud Rahier, Sylvain Marié, Stéphane Girard, Florence Forbes

► To cite this version:

Thibaud Rahier, Sylvain Marié, Stéphane Girard, Florence Forbes. Screening strong pairwise relationships for fast Bayesian network structure learning. 2nd Italian-French Statistics Seminar - IFSS, Sep 2018, Grenoble, France. hal-01941685

HAL Id: hal-01941685

<https://hal.science/hal-01941685v1>

Submitted on 29 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SCREENING STRONG PAIRWISE RELATIONSHIPS FOR FAST BAYESIAN NETWORK STRUCTURE LEARNING

IFSS 2018

Thibaud Rahier, Sylvain Marié, Stéphane Girard, Florence Forbes

September 7, 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

Setting

- $\mathbf{X} = (X_1, \dots, X_n)$: tuple of categorical random variables
- $D = \{(x_1^{(m)}, \dots, x_n^{(m)})\}_{1 \leq m \leq M}$: dataset w/ M i.i.d observations of \mathbf{X}

I-1. BAYESIAN NETWORKS

Setting

- $\mathbf{X} = (X_1, \dots, X_n)$: tuple of categorical random variables
- $D = \{(X_1^{(m)}, \dots, X_n^{(m)})\}_{1 \leq m \leq M}$: dataset w/ M i.i.d observations of \mathbf{X}

Bayesian network: $\mathcal{B} = (G, \Theta)$ where

- $G = (V, A)$: DAG structure with
 - $V = \llbracket 1, n \rrbracket$ 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:

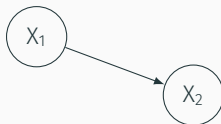
$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | \mathbf{X}_{\pi(i)})$$

- Θ : parameters of the local $P(X_i | \mathbf{X}_{\pi(i)})$

I-2. BAYESIAN NETWORKS: EXAMPLE

$\mathcal{B} = (G, \Theta)$: two node Bayesian network

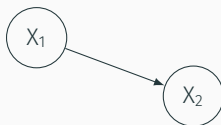
$G = (\{1, 2\}, \{(1, 2)\})$



I-2. BAYESIAN NETWORKS: EXAMPLE

$\mathcal{B} = (G, \Theta)$: two node Bayesian network

$$G = (\{1, 2\}, \{(1, 2)\})$$



$$\Theta = \{\Theta_1, \Theta_2\}$$

- $\Theta_1 = \{\theta_{x_1}\}_{x_1}$ parameters of distribution of X_1 , i.e.:

$$\forall x_1 \in \text{Val}(X_1), \theta_{x_1} = P(X_1 = x_1).$$

- $\Theta_2 = \{\theta_{x_2|x_1}\}_{x_1, x_2}$ parameters of conditional dist. $X_2|X_1$, i.e.:

$$\forall (x_1, x_2) \in \text{Val}(X_1) \times \text{Val}(X_2), \theta_{x_2|x_1} = P(X_2 = x_2 | X_1 = x_1).$$

I-3. BAYESIAN NETWORK STRUCTURE LEARNING

Score&search-based BN structure learning

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

$$\hat{G} \in \underset{G \in \text{DAG}_V}{\text{argmax}} s(G)$$

I-3. BAYESIAN NETWORK STRUCTURE LEARNING

Score&search-based BN structure learning

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

$$\hat{G} \in \underset{G \in \text{DAG}_V}{\text{argmax}} s(G)$$

Some scoring functions

Most scoring functions are based on the log-likelihood $l_D(\Theta)$:

$$l_D(\Theta) = \log(P_{\Theta}(D)) = \sum_{m=1}^M \sum_{i=1}^n \log \left(\theta_{x_i^{(m)} | x_{\pi(i)}^{(m)}} \right)$$

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

$$s_D^{\text{MLL}}(G) = \max_{\Theta \in \Theta_G} l_D(\Theta).$$

I-3. BAYESIAN NETWORK STRUCTURE LEARNING

Score&search-based BN structure learning

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

$$\hat{G} \in \underset{G \in \text{DAG}_V}{\text{argmax}} s(G)$$

Some scoring functions

Most scoring functions are based on the log-likelihood $l_D(\Theta)$:

$$l_D(\Theta) = \log(P_{\Theta}(D)) = \sum_{m=1}^M \sum_{i=1}^n \log \left(\theta_{x_i^{(m)}} | x_{\pi(i)}^{(m)} \right)$$

As the Bayesian information criterion score (BIC):

$$s_D^{\text{BIC}}(G) = \underbrace{\max_{\Theta \in \Theta_G} l_D(\Theta)}_{s_D^{\text{MLL}}(G)} - \frac{\log(M)}{2} d(G),$$

I-3. BAYESIAN NETWORK STRUCTURE LEARNING

Score&search-based BN structure learning

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

$$\hat{G} \in \underset{G \in \text{DAG}_V}{\text{argmax}} s(G)$$

Some scoring functions

Most scoring functions are based on the log-likelihood $l_D(\Theta)$:

$$l_D(\Theta) = \log(P_\Theta(D)) = \sum_{m=1}^M \sum_{i=1}^n \log \left(\theta_{x_i^{(m)}} |x_{\pi(i)}^{(m)} \right)$$

As the Bayesian Dirichlet equivalent score (BDe):

$$s_D^{\text{BDe}}(G) = \log \left(\underbrace{P(G)}_{\text{Structure prior}} \int_{\Theta \in \Theta_G} \underbrace{P(D|\Theta, G)}_{\text{Likelihood}} \underbrace{P(\Theta|G)}_{\text{Dirichlet prior}} d\Theta \right) \propto P(G|D)$$

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

$$s_D^{\text{MLL}}(G) = -M \sum_{i=1}^n H^D(X_i | \mathbf{X}_{\pi(i)})$$

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 **ϵ -quasi deterministic** wrt D iff

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

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 ϵ -**quasi deterministic** wrt D iff

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

Definition: deterministic DAGs

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

$$H^D(X_i|\mathbf{X}_{\pi(i)}) = 0$$

(analogous definition for quasi-deterministic DAGs)

II-3. OPTIMAL BN WITH THE MAXLIKELIHOOD SCORE (1/2)

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 BNSL_{MLL} :

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

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 BNSL_{MLL} :

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

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 BNSL_{MLL} on $\{X_j, j \in R(F)\}$, then $F \cup G_R$ is a solution of BNSL_{MLL} on $\{X_1, \dots, X_n\}$:

$$s_D^{\text{MLL}}(F \cup G_R) = \max_{G \in \text{DAG}_V} s_D^{\text{MLL}}(G)$$

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 a optimal solution to $BNSL_{MLL}$
 - If we can relate subsets of the variables by deterministic trees, solving $BNSL_{MLL}$ 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 $BNSL_{MLL}$
- If we can relate subsets of the variables by deterministic trees, solving $BNSL_{MLL}$ 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**.

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 a optimal solution to $BNSL_{MLL}$
- If we can relate subsets of the variables by deterministic trees, solving $BNSL_{MLL}$ 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, ϵ , sota-BNSL

- 1: Compute F_ϵ by running **qd-screening** with D and ϵ
- 2: Identify $R(F_\epsilon) = \{i \in \llbracket 1, n \rrbracket \mid \pi^{F_\epsilon}(i) = \emptyset\}$, the set of F_ϵ 's roots.
- 3: Compute $G_{R(F_\epsilon)}^*$ by running sota-BNSL on $\mathbf{X}_{R(F_\epsilon)}$
- 4: $G_\epsilon^* \leftarrow F_\epsilon \cup G_{R(F_\epsilon)}^*$

Output: G_ϵ^*

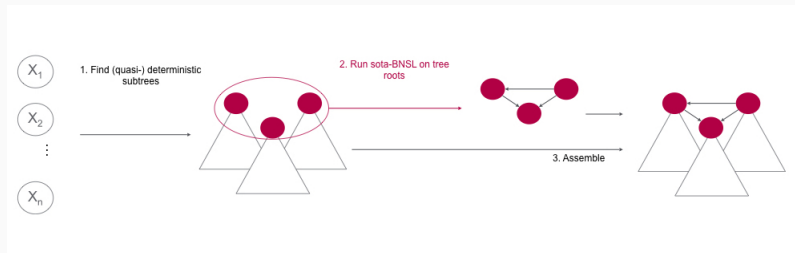
III-2. BNSL WITH QD-SCREENING: ALGORITHM

Algorithm 2 Bayesian network structure learning with quasi deterministic screening (qds-BNSL)

Input: D , ϵ , sota-BNSL

- 1: Compute F_ϵ by running **qd-screening** with D and ϵ
- 2: Identify $R(F_\epsilon) = \{i \in \llbracket 1, n \rrbracket \mid \pi^{F_\epsilon}(i) = \emptyset\}$, the set of F_ϵ '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_ϵ^*



Algorithm 3 Bayesian network structure learning with quasi deterministic screening (qds-BNSL)

Input: D, ϵ , sota-BNSL

- 1: Compute F_ϵ by running **qd-screening** with D and ϵ
- 2: Identify $R(F_\epsilon) = \{i \in \llbracket 1, n \rrbracket \mid \pi^{F_\epsilon}(i) = \emptyset\}$, the set of F_ϵ '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_ϵ^*

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 $|R(F_\epsilon)| < n$ (Rahier et al., 2018)

I - Bayesian network structure learning

II - Determinism and Bayesian networks

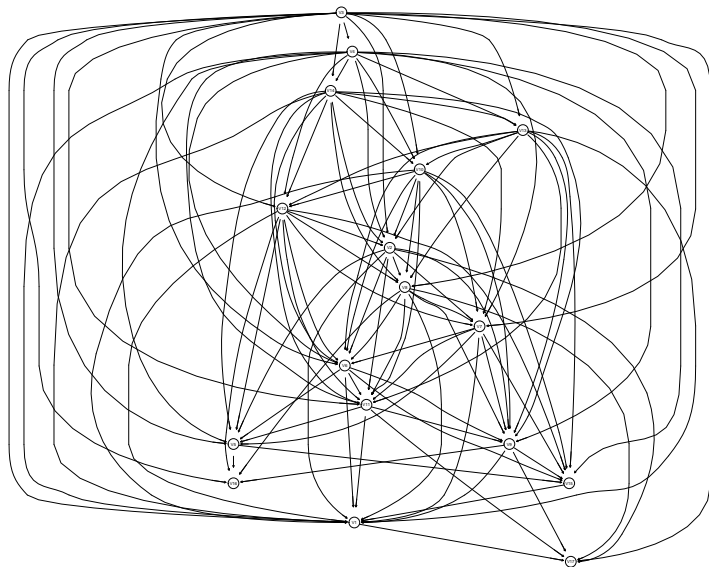
III - Structure learning with (quasi-)determinism screening

IV - Experiments

V - Discussion

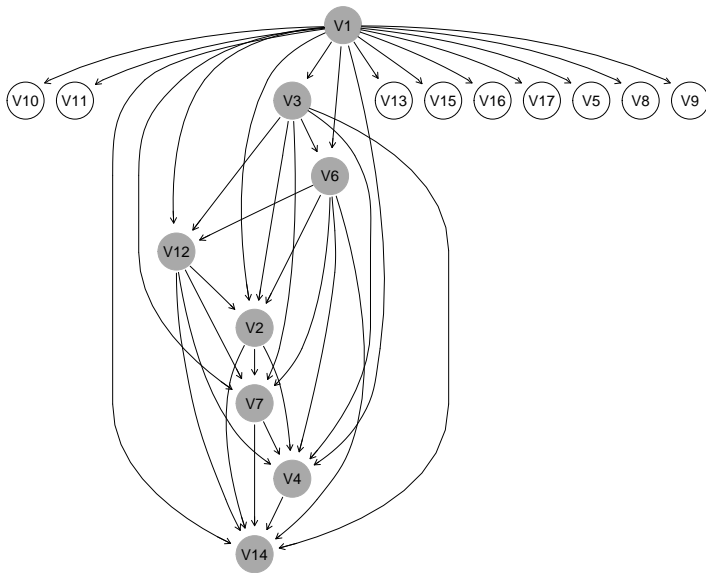
IV-1. BAYESIAN NETWORKS LEARNT ON THE MSNBC DATASET: BASELINE

BN learnt on dataset 'msnbc' with sota-BNSL

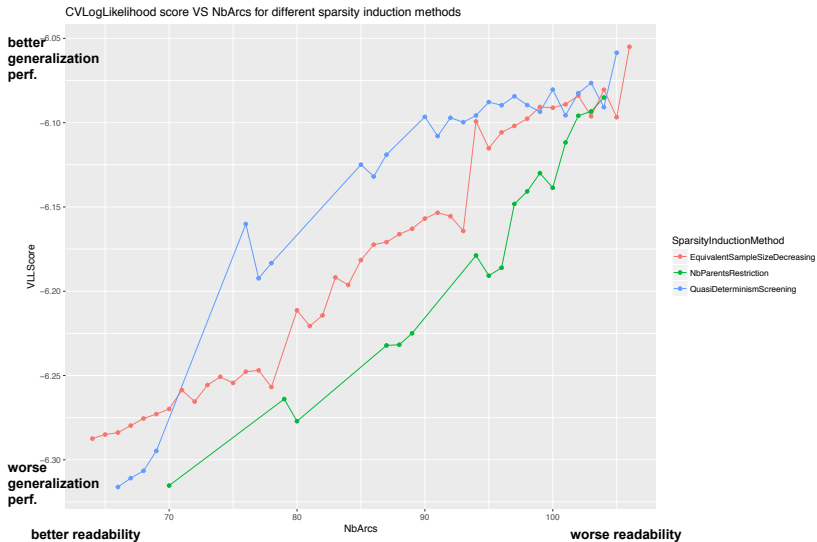


IV-1. BAYESIAN NETWORKS LEARNT ON THE MSNBC DATASET: QDS

BN learnt on dataset 'msnbc' with qds-BNSL (eps_0.5)



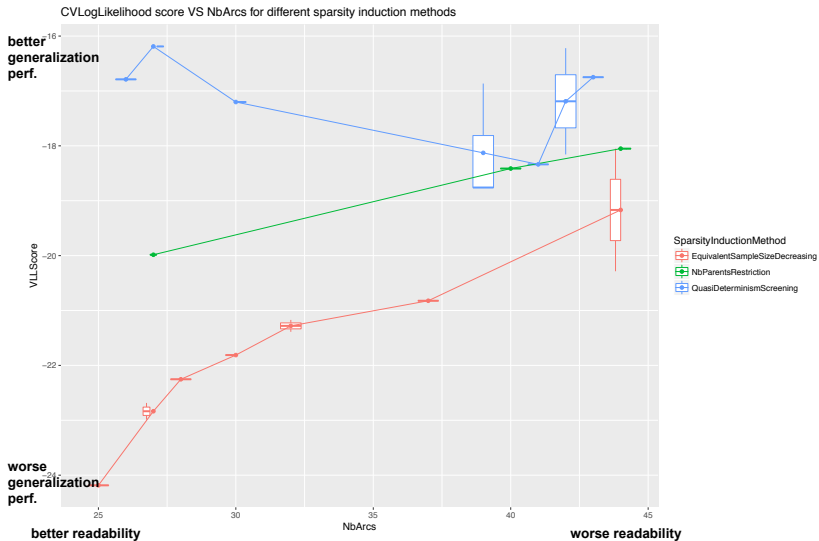
IV-2. PERFORMANCE/READABILITY TRADEOFF - MSNBC DATASET



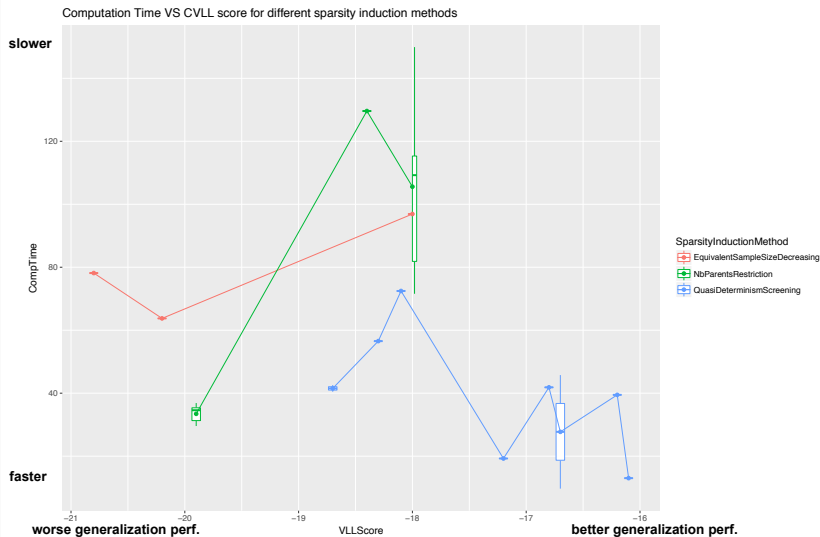
IV-3. PERFORMANCE/COMPUTATION TIME TRADEOFF - MSNBC DATASET



IV-4. PERFORMANCE / READABILITY TRADEOFF - PIU DATASET



IV-5. PERFORMANCE/COMPUTATION TIME TRADEOFF - PIU DATASET



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

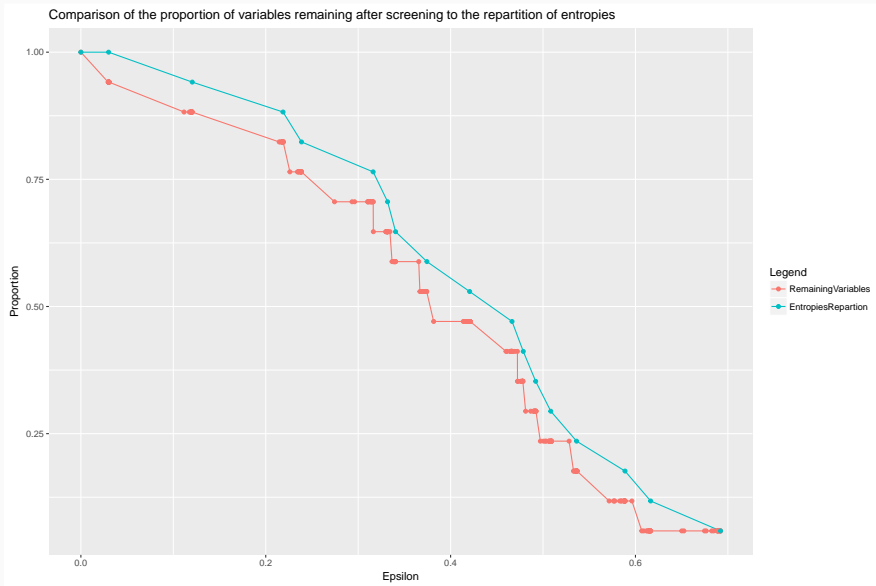
In the future we plan to

- Search for **guarantees** of qds-BNSL wrt scores as BIC, BDe or CVLL
- Look for a **criteria** that enables us to choose ϵ in a principled way

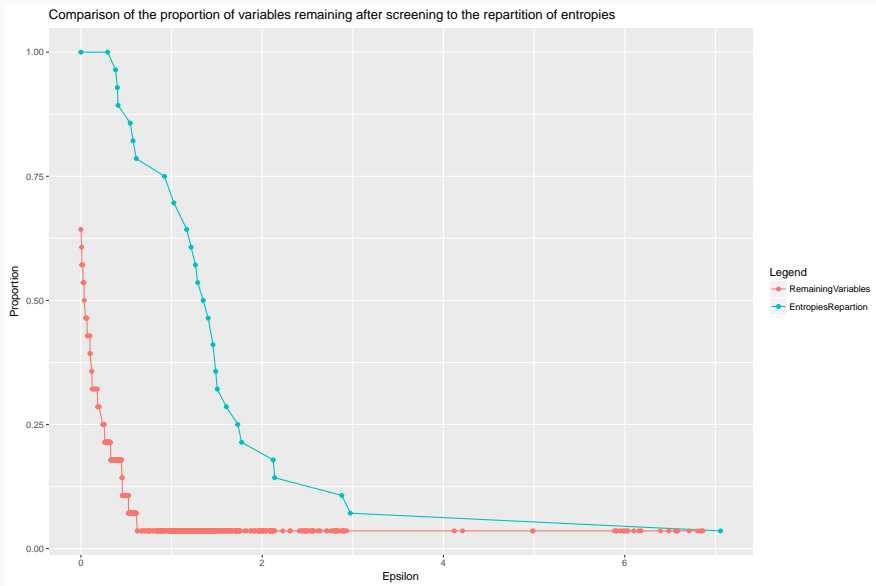
THANK YOU

MORE RESULTS

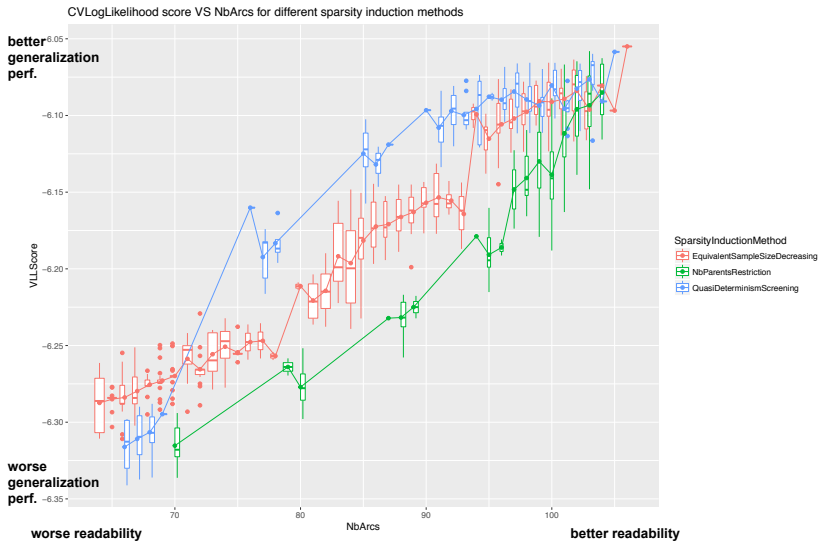
V-2. CANDIDATE CRITERION FOR CHOICE OF ϵ - MSNBC DATASET



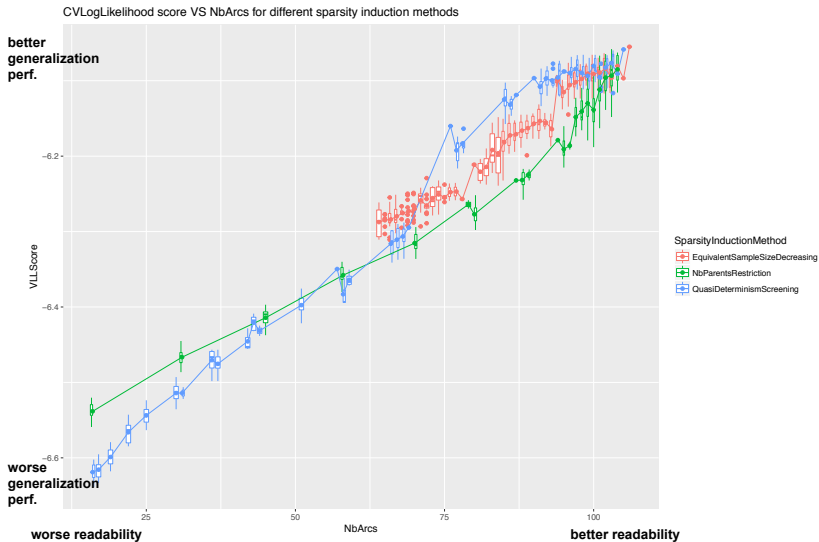
V-3. CANDIDATE CRITERION FOR CHOICE OF ϵ - PIU DATASET



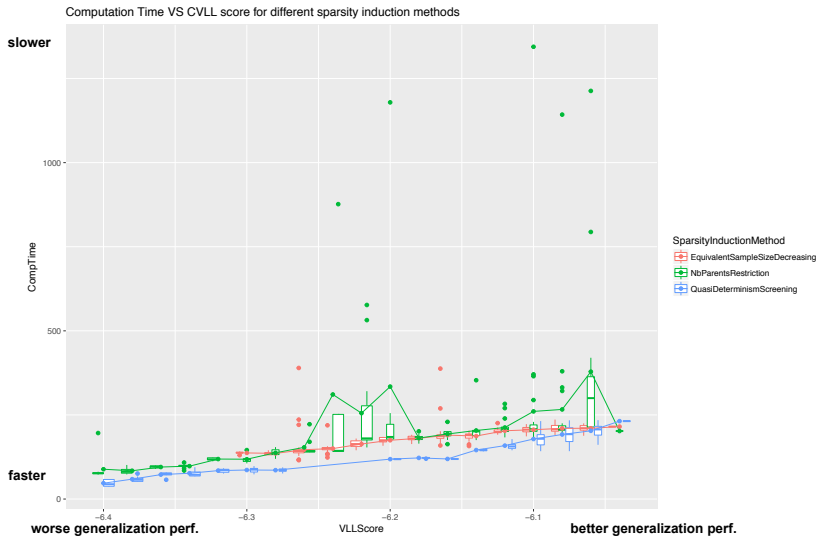
APP 1. PERFORMANCE/READABILITY TRADEOFF - MSNBC DATASET (1/2)



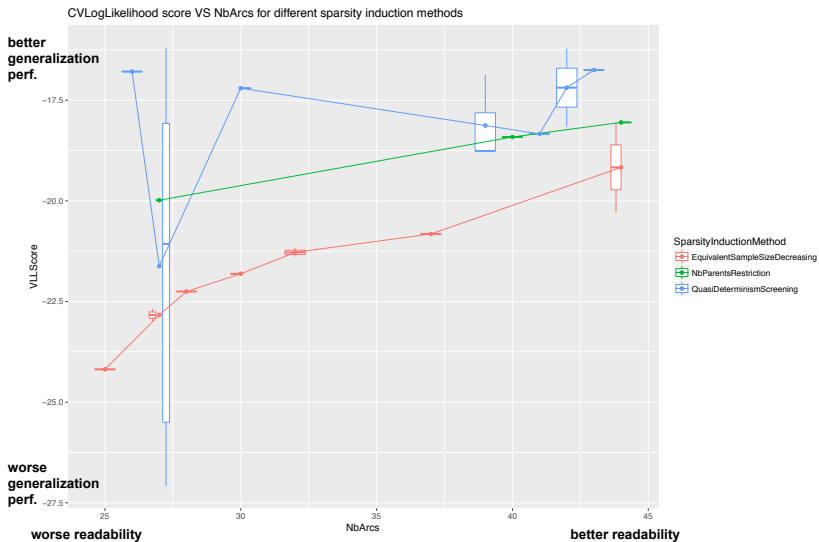
APP 1. PERFORMANCE/READABILITY TRADEOFF - MSNBC DATASET (2/2)



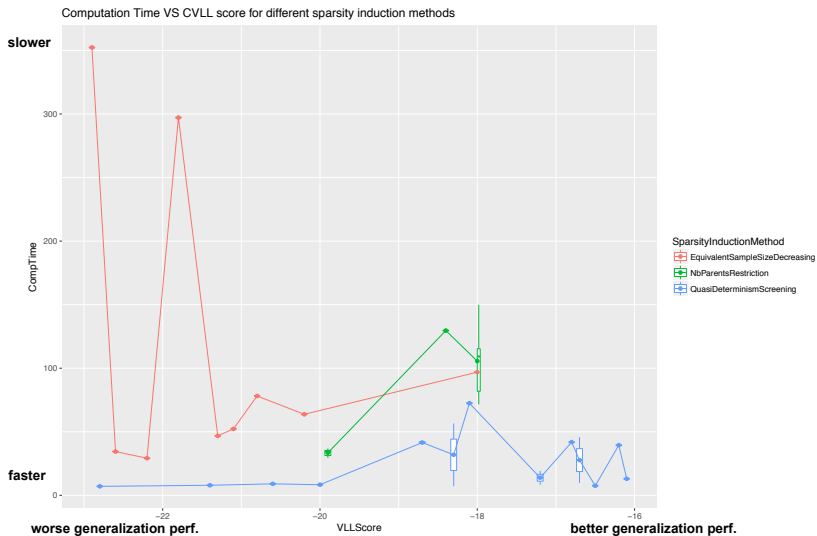
APP 2. PERFORMANCE/TIME TRADEOFF - MSNBC DATASET



APP 3. PERFORMANCE/READABILITY TRADEOFF - PIU DATASET



APP 4. PERFORMANCE/TIME TRADEOFF - PIU DATASET

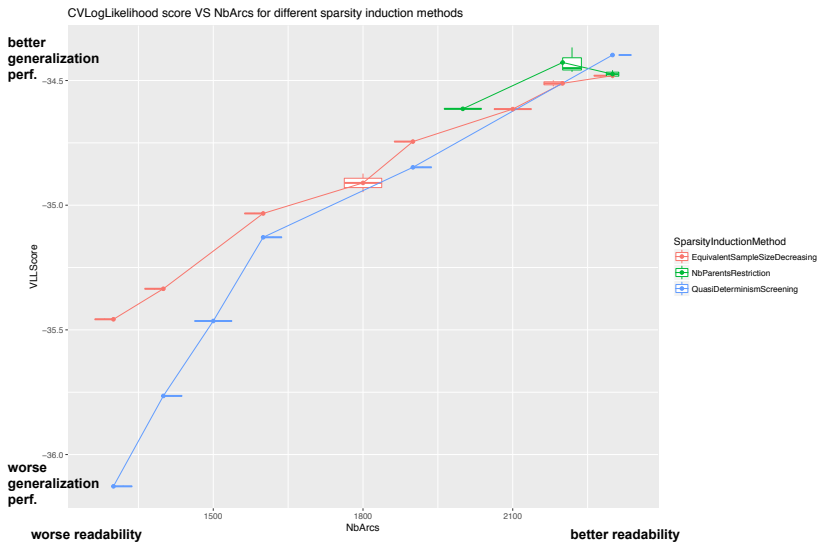


Algorithm 4 Quasi-determinism screening (qds)

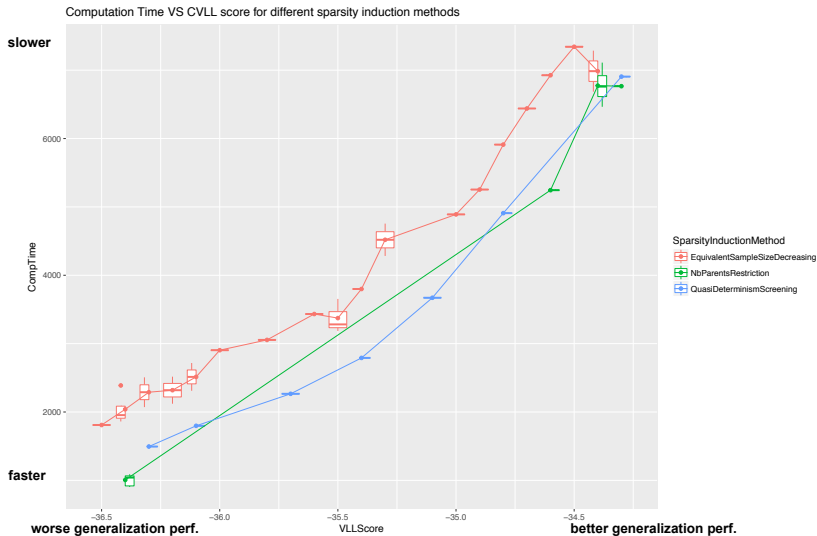
Input: D, ϵ

- 1: Compute empirical cond. entropy matrix $\mathbb{H}^D = (\mathbb{H}^D(X_i|X_j))_{1 \leq i, j \leq n}$
 - 2: **for** $i = 1$ to n **do**
 - 3: compute $\pi_\epsilon(i) = \{j \in \llbracket 1, n \rrbracket \setminus \{i\} \mid \mathbb{H}_{ij}^D \leq \epsilon\}$
 - 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** $\mathbb{H}_{ij}^D \leq \mathbb{H}_{ji}^D$ **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: $\pi_\epsilon^*(i) \leftarrow \underset{j \in \pi_\epsilon(i)}{\operatorname{argmin}} |\operatorname{Val}(X_j)|$
 - 10: Compute forest $F_\epsilon = (V_{F_\epsilon}, A_{F_\epsilon})$, where
 - $V_{F_\epsilon} = \llbracket 1, n \rrbracket$
 - $A_{F_\epsilon} = \{(\pi_\epsilon^*(i), i) \mid i \in \llbracket 1, n \rrbracket \text{ s.t. } \pi_\epsilon^*(i) \neq \emptyset\}$
- Output:** F_ϵ

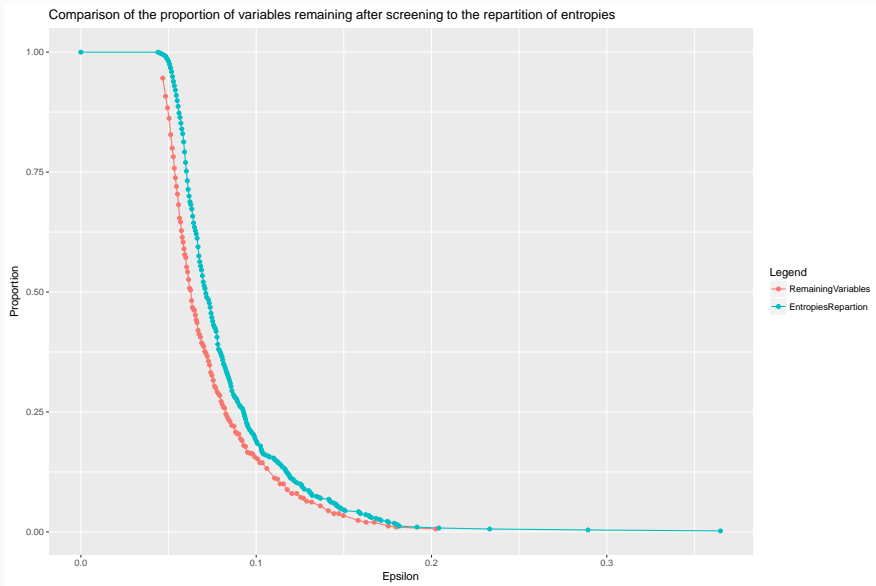
APP 6. PERFORMANCE/READABILITY TRADEOFF - BOOK DATASET



APP 7. PERFORMANCE/COMPUTATION TIME TRADEOFF - BOOK DATASET



APP 8.. CANDIDATE CRITERION FOR CHOICE OF ϵ - BOOK DATASET



Thibaud Rahier, Sylvain Marié, Stéphane Girard, and Florence Forbes. Fast bayesian network structure learning using quasi-determinism screening. In JFRB 2018-9èmes Journées Francophones sur les Réseaux Bayésiens et les Modèles Graphiques Probabilistes, pages 14–24, 2018.