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

OUTLINE

I - Bayesian network structure learning

II - Determinism and Bayesian networks

III - Structure learning with (quasi-)determinism screening

IV - Experiments

V - Discussion

Setting

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

Setting

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

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

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

  Factorization of the joint distribution:

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

- θ: parameters of the local $P(X_i | \mathbb{X}_{\pi_i})$

Score&search-based BN structrure learning

For a scoring function $s : DAG_V \to \mathbb{R}$, $BNSL_s$ comes down to:

$$\hat{G} \in \underset{G \in DAG_V}{\operatorname{argmax}}\, s(G)$$

Score&search-based BN structrure learning

For a scoring function $s : DAG_V \to \mathbb{R}$, $BNSL_s$ comes down to:

$$\hat{G} \in \underset{G \in DAG_V}{\operatorname{argmax}}\, s(G)$$

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):

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

In practice, we rather use regularized scores such as BIC, AIC or BDe

OUTLINE

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

$$s^{MLL}(G : D) = -M \sum_{i=1}^{n} H^{D}(X_i|\mathbf{X}_{\pi_i})$$

Definitions: determinism and quasi-determinism

The relationship $X_i \to X_j$ is deterministic wrt D iff

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

The relationship $X_i \to X_j$ 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 $T \in DAG_V$ is a deterministic tree (single-parented DAG) wrt D then T is a solution of $BNSL_{MLL}$:

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

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

Proposition 1: Deterministic trees and the MLL score

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

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

Proposition 2: Deterministic forests and the MLL score

Let $F \in DAG_V$ 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 $\{X_1, \ldots, X_n\}$:

$$s^{MLL}(F \cup G_R : D) = \max_{G \in DAG_V} s^{MLL}(G : D)$$

OUTLINE

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

$\longrightarrow$ 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

$\longrightarrow$ 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.

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

$\longrightarrow$ 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
$\longrightarrow$ 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 $R(F_\epsilon) = \{i \in [\![1, n]\!] \mid \pi^{F_\epsilon}(i) = \emptyset\}$, 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 2 Bayesian network structure learning with quasi deterministic screening (qds-BNSL)

Input: D, ϵ, sota-BNSL

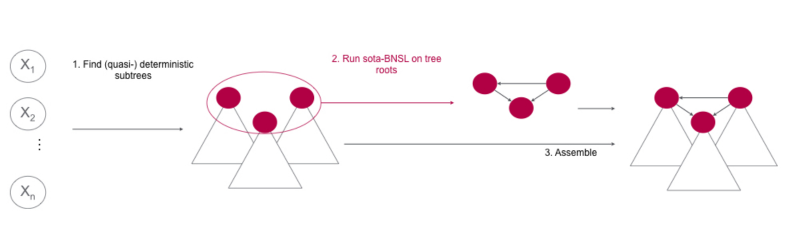
1: Compute F_ϵ by running **qd-screening** with D and ϵ

2: Identify $R(F_\epsilon) = \{i \in [\![1, n]\!] \mid \pi^{F_\epsilon}(i) = \emptyset\}$, 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 $R(F_\epsilon) = \{i \in [\![1, n]\!] \mid \pi^{F_\epsilon}(i) = \emptyset\}$, the set of F_ϵ's roots.
3: Compute $G^*_{R(F_\epsilon)}$ by running sota-BNSL on $\mathbf{X}_{R(F_\epsilon)}$
4: $G^*_\epsilon \leftarrow F_\epsilon \cup G^*_{R(F_\epsilon)}$

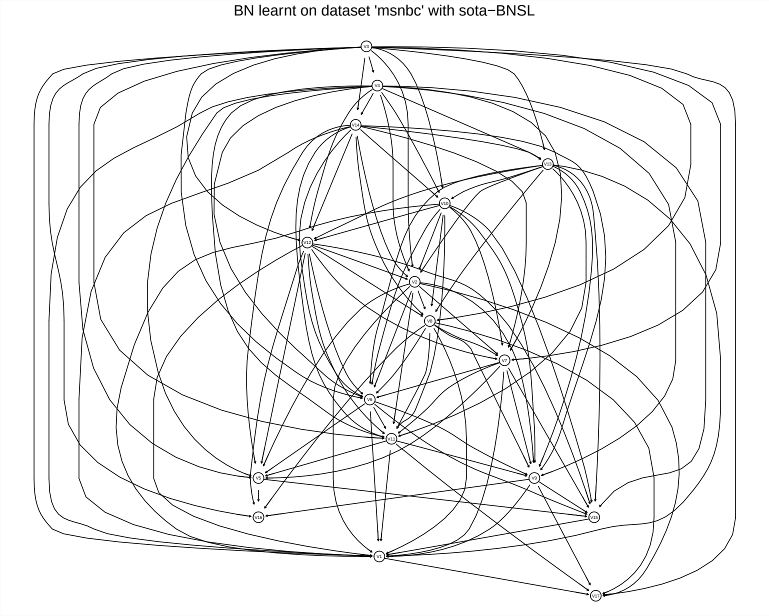
Output: G^*_ϵ

Complexity

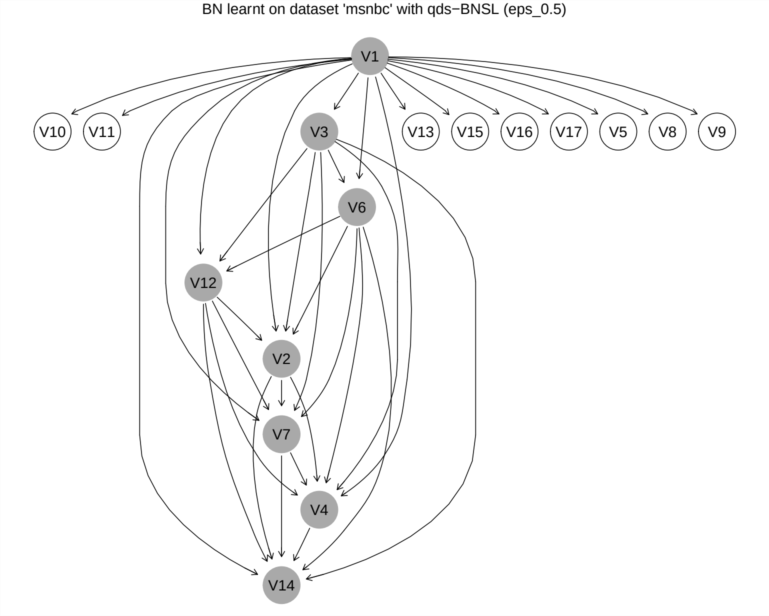
- qd-screening: $O(n^2)$
- qds-BNSL: calls sota-BNSL on $|R(F_\epsilon)| \leq n$ variables (exact BNSL: $O(2^p)$, heuristics are very time-intensive as well)

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

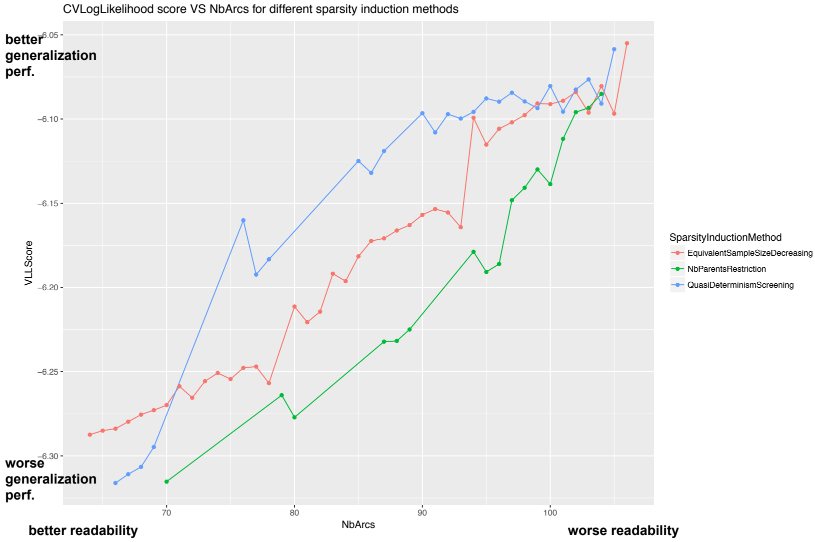
OUTLINE


BN learnt on dataset 'msnbc' with sota−BNSL
V3
V4
V14
V13
V10
V12
V2
V8
V7
V6
V11
V5
V9
V16
V15
V1
V17

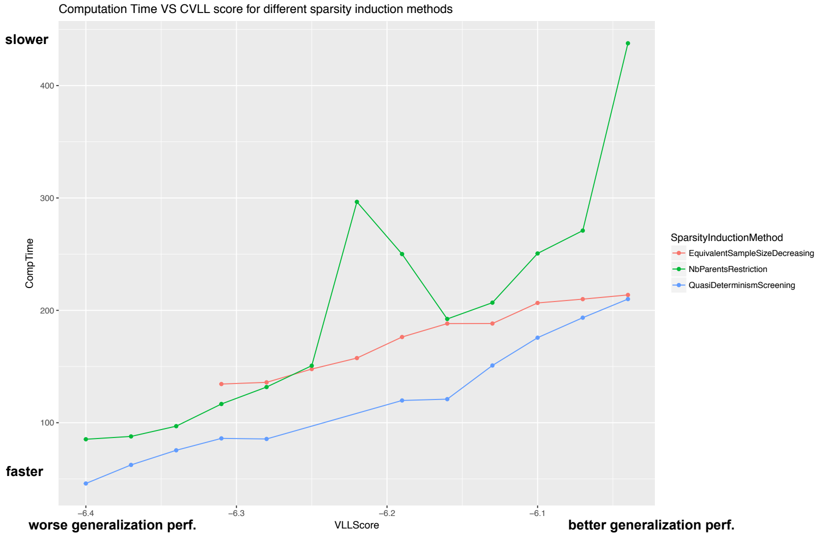
IV-1. BAYESIAN NETWORKS LEARNT ON THE MSNBC DATASET: QDS



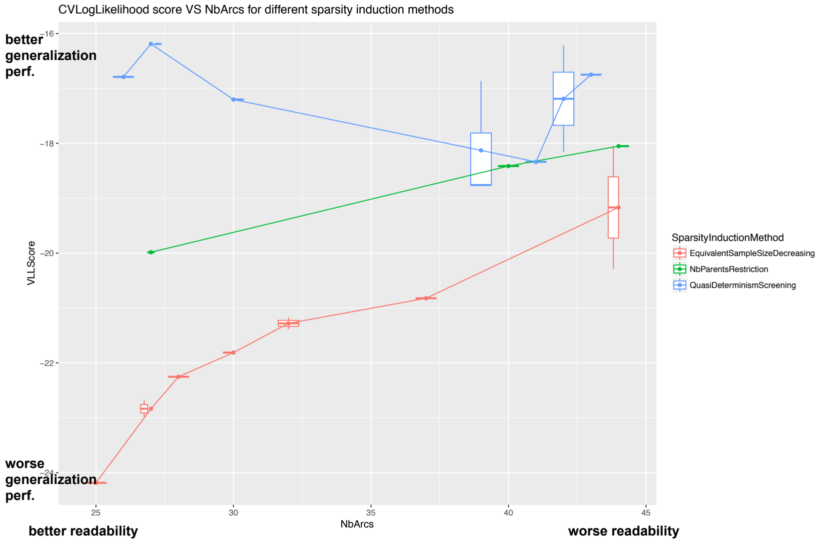
IV-2. PERFORMANCE/READABILITY TRADEOFF - MSNBC DATASET



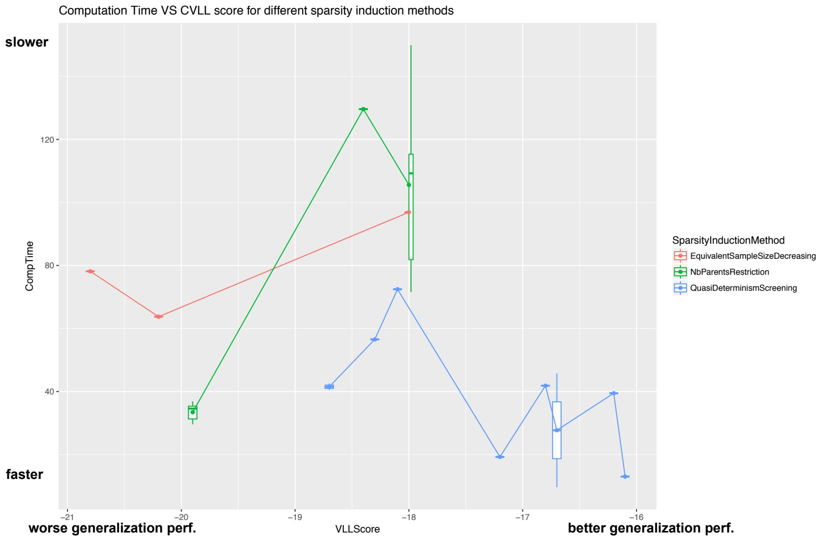
IV-3. PERFORMANCE/COMPUTATION TIME TRADEOFF - MSNBC DATASET



IV-4. PERFORMANCE/READABILITY TRADEOFF - PIU DATASET



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



OUTLINE

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

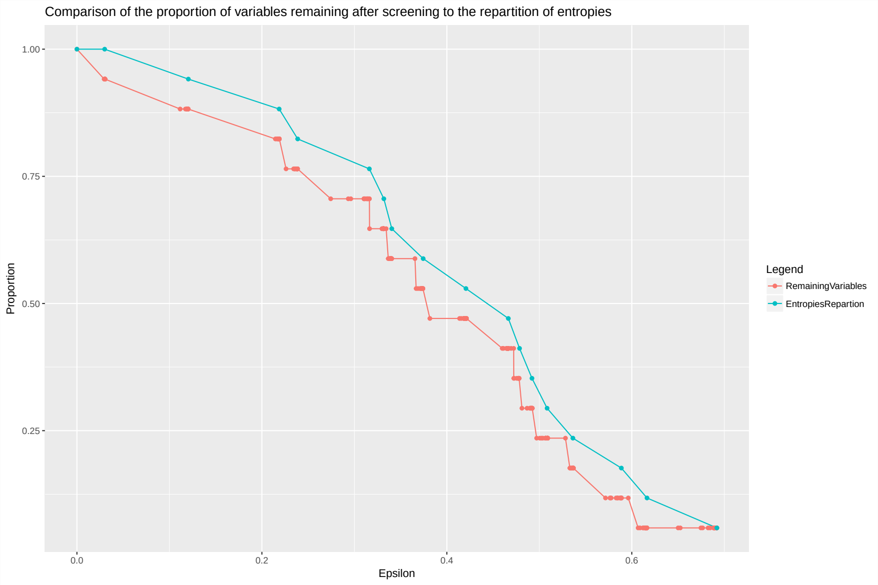
However these properties depend highly on the dataset

Perspectives

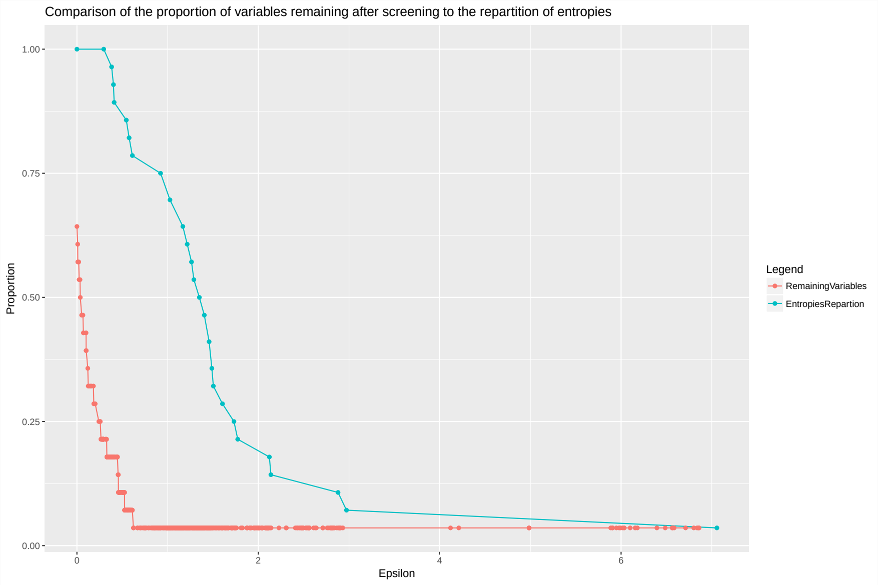
In the future we plan to

- 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

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



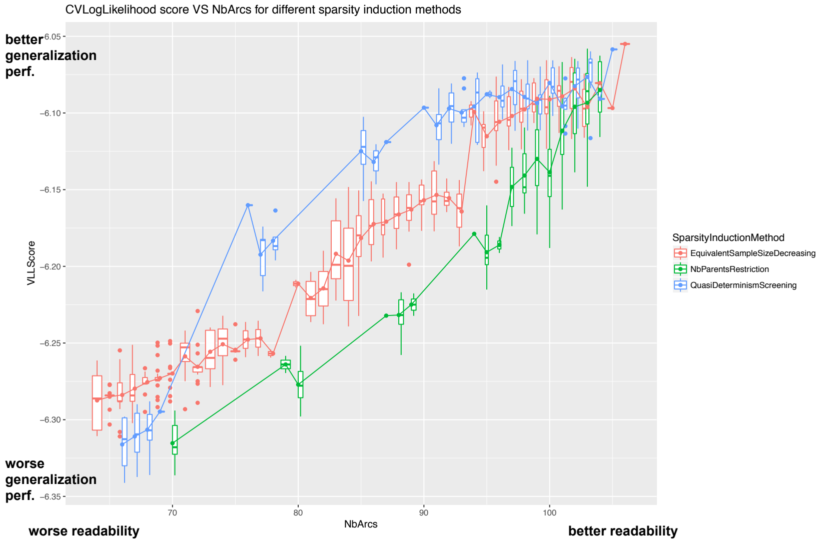
V-3. CANDIDATE CRITERION FOR CHOICE OF ϵ - PIU DATASET



Thank you

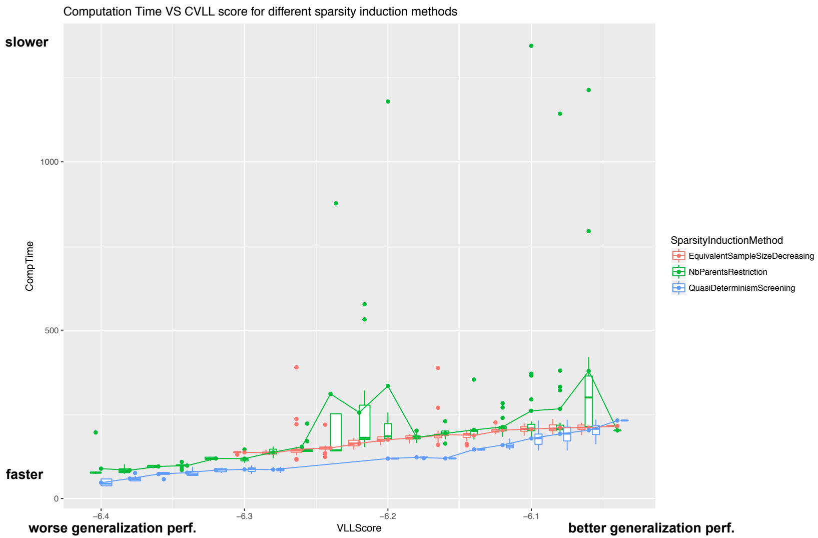
More results

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

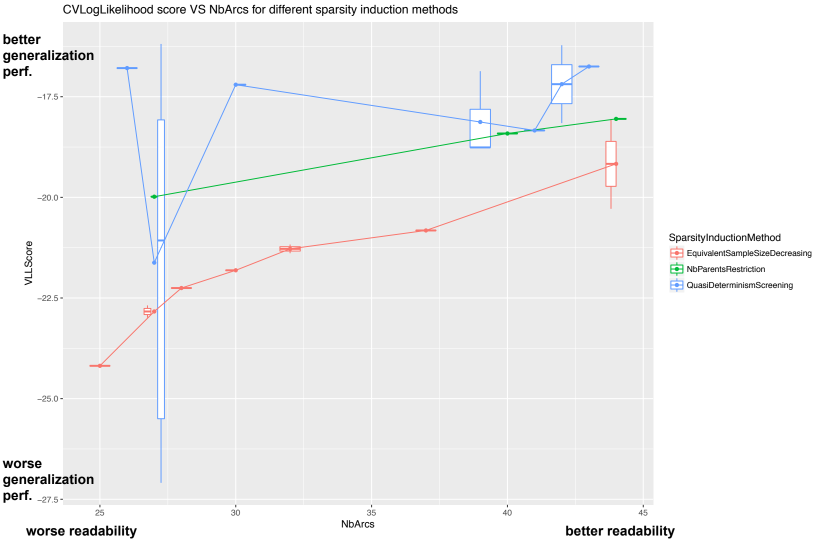


APP 1. PERFORMANCE/READABILITY TRADEOFF - MSNBC DATASET (2/2)

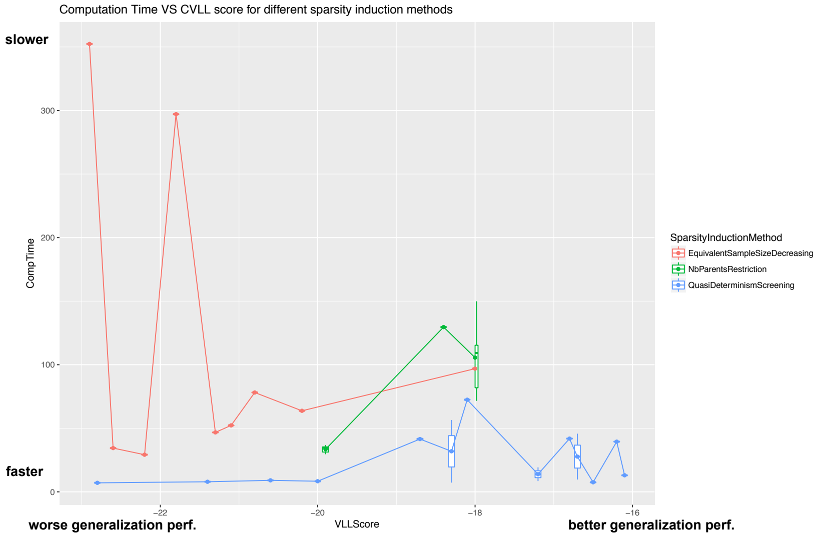
App 2. Performance/time tradeoff - MSNBC dataset



APP 3. PERFORMANCE/READABILITY TRADEOFF - PIU DATASET



APP 4. PERFORMANCE/TIME TRADEOFF - PIU DATASET



Algorithm 4 Quasi-determinism screening (qds)

Input: D , ϵ

Compute empirical cond. entropy matrix $\mathbb{H}^D = \left(H^D(X_i|X_j)\right)_{1\leq i,j\leq n}$

for $i = 1$ to n **do**

  compute $\pi_\epsilon(i) = \{j \in [\![1, n]\!] \setminus \{i\} \mid \mathbb{H}^D_{ij} \leq \epsilon\}$

for $i = 1$ to n **do**

  if $\exists j \in \pi_\epsilon(i)$ s.t. $i \in \pi_\epsilon(j)$ **then**

    if $\mathbb{H}^D_{ij} \leq \mathbb{H}^D_{ji}$ **then** $\pi_\epsilon(j) \leftarrow \pi_\epsilon(j) \setminus \{i\}$

    else $\pi_\epsilon(i) \leftarrow \pi_\epsilon(i) \setminus \{j\}$

for $i = 1$ to n **do**

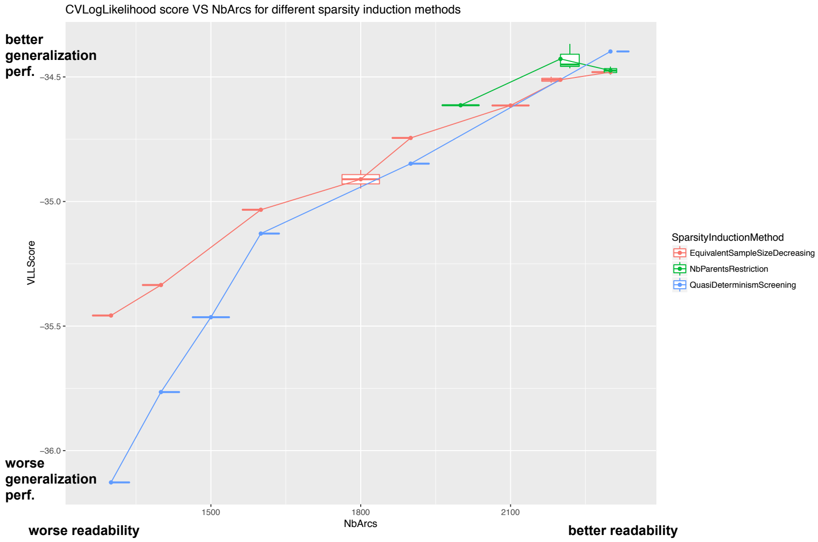
  $\pi^*_\epsilon(i) \leftarrow \underset{j\in\pi_\epsilon(i)}{\operatorname{argmin}} |Val(X_j)|$

Compute forest $F_\epsilon = (V_{F_\epsilon}, A_{F_\epsilon})$, where

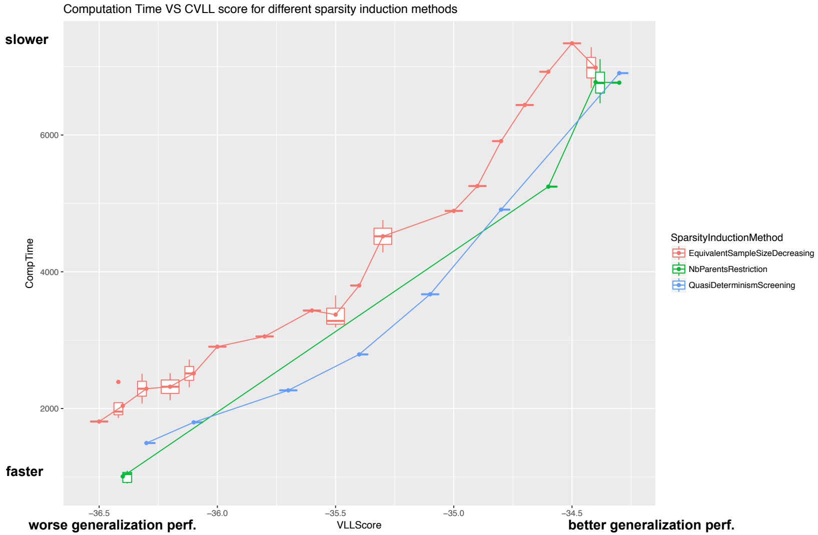
  $V_{F_\epsilon} = [\![1, n]\!]$

  $A_{F_\epsilon} = \{(\pi^*_\epsilon(i), i) \mid i \in [\![1, n]\!] \text{ s.t. } \pi^*_\epsilon(i) \neq \emptyset\}$

Output: F_ϵ


CVLogLikelihood score VS NbArcs for different sparsity induction methods
better generalization perf.
worse generalization perf.
worse readability
better readability
VLLScore
-34.5
-35.0
-35.5
-36.0
1500
1800
2100
NbArcs
SparsityInductionMethod
EquivalentSampleSizeDecreasing
NbParentsRestriction
QuasiDeterminismScreening

APP 7. PERFORMANCE/COMPUTATION TIME TRADEOFF - BOOK DATASET



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