Bayesian Network Models for Continuous-Time and Structured Data

Marco Scutari
scutari@bnlearn.com

Dalle Molle Institute for
Artificial Intelligence (IDSIA)

September 7, 2022

→ Bayesian Networks: Definition and Assumptions

Continuous-Time Bayesian Networks

Bayesian Networks for Structured Data

Future Directions

A Graph and a Probability Distribution

A Bayesian network (BN) is defined by:

- a network structure, a directed acyclic graph $\mathcal{G}$ in which each node corresponds to a random variable X_i;
- a global probability distribution $\mathbf{X}$ with parameters Θ, which can be factorised into smaller local probability distributions according to the arcs present in $\mathcal{G}$.

The main role of the network structure is to express the conditional independence relationships among the variables in the model through graphical separation, thus specifying the factorisation of the global distribution:

$$\mathrm{P}(\mathbf{X}) = \prod_{i=1}^{N} \mathrm{P}(X_i \mid \Pi_{X_i}; \Theta_{X_i}) \quad \text{where} \quad \Pi_{X_i} = \{\text{parents of } X_i \text{ in } \mathcal{G}\}\,.$$

Learning a BN $\mathcal{B} = (\mathcal{G}, \Theta)$ from a data set $\mathcal{D}$ involves two steps:

$$\underbrace{\mathrm{P}(\mathcal{B} \mid \mathcal{D}) = \mathrm{P}(\mathcal{G}, \Theta \mid \mathcal{D})}_{\text{learning}} = \underbrace{\mathrm{P}(\mathcal{G} \mid \mathcal{D})}_{\text{structure learning}} \cdot \underbrace{\mathrm{P}(\Theta \mid \mathcal{G}, \mathcal{D})}_{\text{parameter learning}}.$$

Structure learning consists in finding the DAG with the best

$$\mathrm{P}(\mathcal{G} \mid \mathcal{D}) \propto \underbrace{\mathrm{P}(\mathcal{G})}_{\text{graph prior}} \cdot \underbrace{\mathrm{P}(\mathcal{D} \mid \mathcal{G})}_{\text{marginal likelihood}} = \mathrm{P}(\mathcal{G}) \int \mathrm{P}(\mathcal{D} \mid \mathcal{G}, \Theta)\, \mathrm{P}(\Theta \mid \mathcal{G})\, d\Theta$$

which is known as score-based learning [6]. The alternative, constraint-based learning, uses tests following Pearl's work on causality [14]:

$$\underbrace{X_i \perp\!\!\!\perp_P X_j \mid \mathbf{S}_{X_i, X_j}}_{\text{conditional independence}} \Longrightarrow \underbrace{X_i \perp\!\!\!\perp_G X_j \mid \mathbf{S}_{X_i, X_j}}_{\text{graphical separation}}.$$

Parameter learning consists in estimating the parameters $\Theta_{X_i} \mid \Pi_{X_i}$.

The Classic Definition and Modern Extensions

What are we assuming when trying to learn a BN? Typically that:

- observations are independent and there are no missing values;
- all variables are observed, that is, there are no latent variables introducing confounding in the model;
- we measure probabilistic associations (or rather, independencies) and we cannot necessarily interpret them as causal.

What happens if we relax these assumptions? Many extensions suddenly become possible, see [11] for a recent review. In this talk we will discuss:

- Learning BNs from continuous-time dynamic data [4].
- Learning BNs from heterogeneous data that are the collation of multiple related data sets [1].

We will not discuss learning BNs from incomplete data, but we are making progress on that front as well [3].

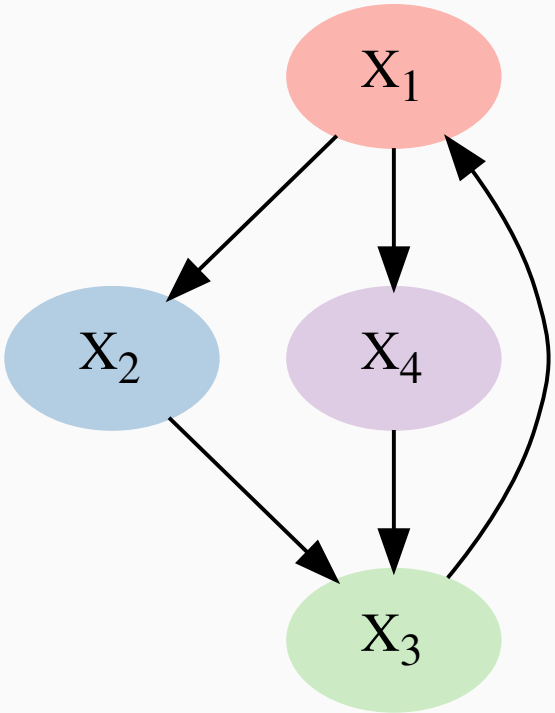
- ✓ Bayesian Networks: Definition and Assumptions
- → Continuous-Time Bayesian Networks
- Bayesian Networks for Structured Data
- Future Directions

Continuous-Time Bayesian Networks

Continuous-Time BNs (CTBNs) are a framework for modelling finite-state, continuous-time processes. Their graphical representation allows for natural, cyclic dependency graphs without having to specify a temporal granularity [9].

A CTBN consists of two components:

- A directed graph encoding conditional independencies.
- A conditional intensity matrix (CIM) $\mathbf{Q}_{X_i \mid \mathbf{u}}$ describing the evolution process of a variable with the parameters
  - $\mathbf{q}_{X_i}$: a set of intensities parameterising the exponential distributions over when the next transition occurs.
  - $\boldsymbol{\theta}_{X_i}$: a set of probabilities parameterising the distribution over where the state transitions.



Constraint-Based Structure Learning?

Score-based learning was covered by Nodelman [9] in his original work on CTBNs. For constraint-based structure learning we need a new definition of conditional independence [4]:

> Let $\mathcal{N}$ be a CTBN with a graph $\mathcal{G}$ over $\mathbf{X}$. We say that $X_i \perp\!\!\!\perp X_j \mid \mathbf{S}_{X_i,X_j}$ if $\mathbf{Q}_{X_i \mid x,\mathbf{s}} = \mathbf{Q}_{X_i \mid \mathbf{s}}$ for all values x, s of X_j and $\mathbf{S}_{X_i,X_j}$.

Note that conditional independence is not symmetric in CTBNs! To test it we need to test two separate hypotheses:

- Time To Transition: independence of the waiting times ($\mathbf{q}_{X_i}$), tested with an F test to compare their exponential distributions.
- State-to-State Transition: independence of the transitions ($\boldsymbol{\theta}_{X_i}$), tested with a two-sample χ^2 test or a Kolmogorov-Smirnov test.

We test time-to-transition hypothesis first and then, if the null is rejected, the state-to-state hypotheses. If both nulls are rejected, X_i and X_j are conditionally independent.

Time to Transition [2]: given the exponential waiting times $q_{x|\mathbf{s}}, q_{x|y,\mathbf{s}}$,

$$H_0 : \frac{q_{x|\mathbf{s}}}{q_{x|y,\mathbf{s}}} = 1 \qquad \text{with null } F_{r_a, r_b}$$

where $r_a = \sum_{x' \in X_i} M_{xx'|y,\mathbf{s}}$ and $r_b = \sum_{x' \in X_i} M_{xx'|\mathbf{s}}$.

State-to-State Transition [8]: given $\theta_{x|\mathbf{s}}, \theta_{x|y,\mathbf{s}}$,

$$H_0 : \theta_{x|\mathbf{s}} = \theta_{x|y,\mathbf{s}} \quad \text{with null} \quad \chi^2 = \sum_{x' \in X_i} \frac{(K \cdot M_{xx'|y,\mathbf{s}} - L \cdot M_{xx'|\mathbf{s}})^2}{M_{xx'|\mathbf{s}} + M_{xx'|y,\mathbf{s}}}$$

where $K = \sqrt{\frac{\sum_{i=1}^{k} M_{xx'|\mathbf{s}}}{\sum_{i=1}^{k} M_{xx'|y,\mathbf{s}}}}$ and $L = \frac{1}{K}$.

We reject the (conditional) independence between the two nodes if at least null hypothesis is rejected.

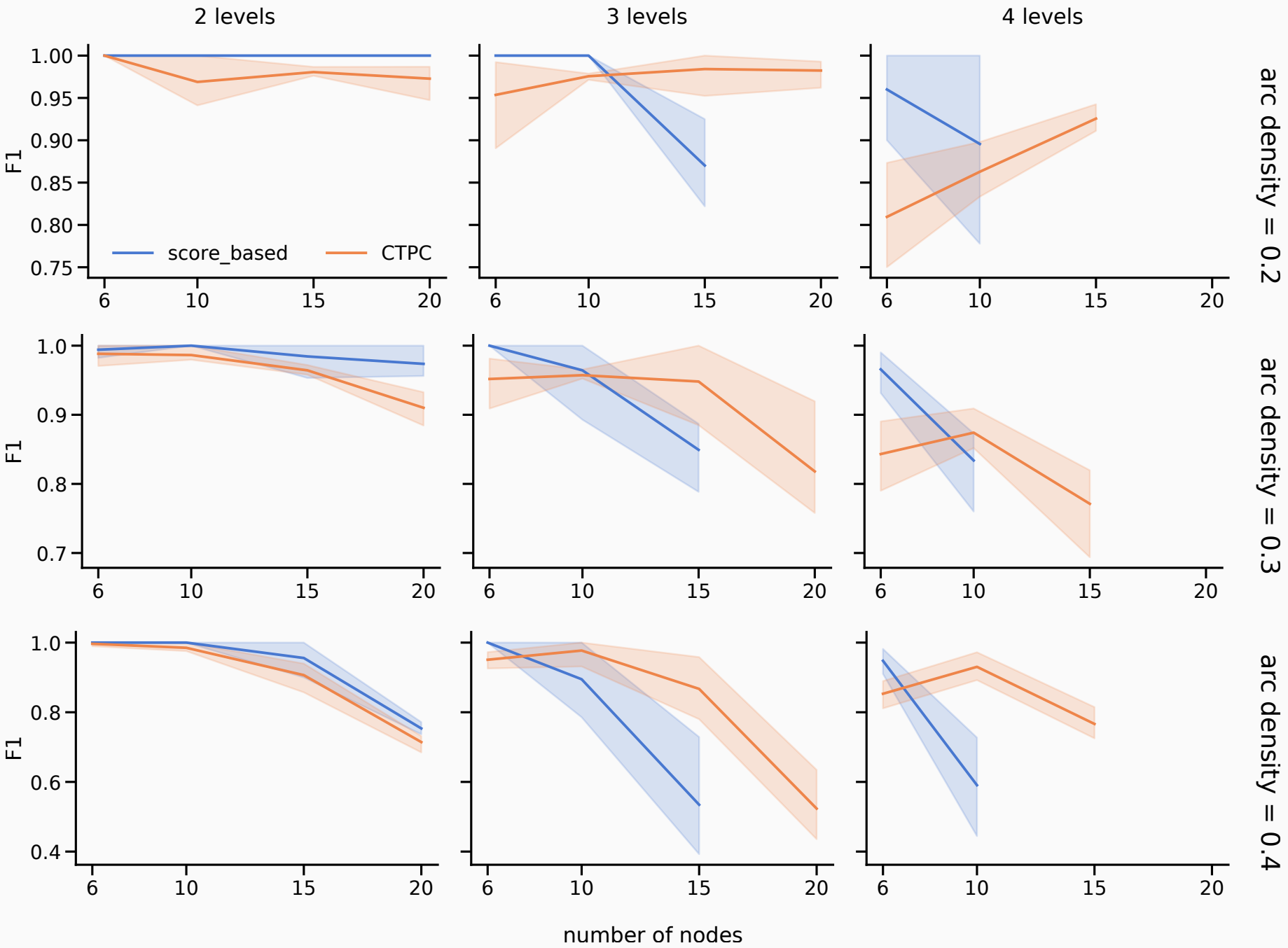
A PC Algorithm for Continuous-Time Bayesian Networks

Given how different is the definition of conditional independence, we need to adapt the PC algorithm [5] to match.

1. Form a complete directed graph $\mathcal{G}$ over $\mathbf{X}$.
2. For each variable X_i:
   - 2.1 Set $\mathbf{U} = \{X_j \in \mathbf{X} : X_j \to X_i\}$, the current parent set.
   - 2.2 For increasing values $b = 0, \ldots, |\mathbf{U}|$:
     - 2.2.1 For each $X_j \in \mathbf{U}$, test $X_i \perp\!\!\!\perp X_j \mid \mathbf{S}_{X_i, X_j}$ for all possible subsets of size b of $\mathbf{U} \setminus X_j$.
     - 2.2.2 As soon as $X_i \perp\!\!\!\perp X_j \mid \mathbf{S}_{X_i, X_j}$ for some $\mathbf{S}_{X_i, X_j}$, remove $X_j \to X_i$ from $\mathcal{G}$ and X_j from $\mathbf{U}$.
3. Return $\mathcal{G}$.

We call this the Continuous-Time PC (CTPC) algorithm [4]. It has better structural reconstruction accuracy than the score-based approach in [9], but both approaches are slow: they are only practical for less than 20 variables.

CTPC Versus Score-Based Learning



- ✓ Bayesian Networks: Definition and Assumptions
- ✓ Continuous-Time Bayesian Networks
- → **Bayesian Networks for Structured Data**
- Future Directions

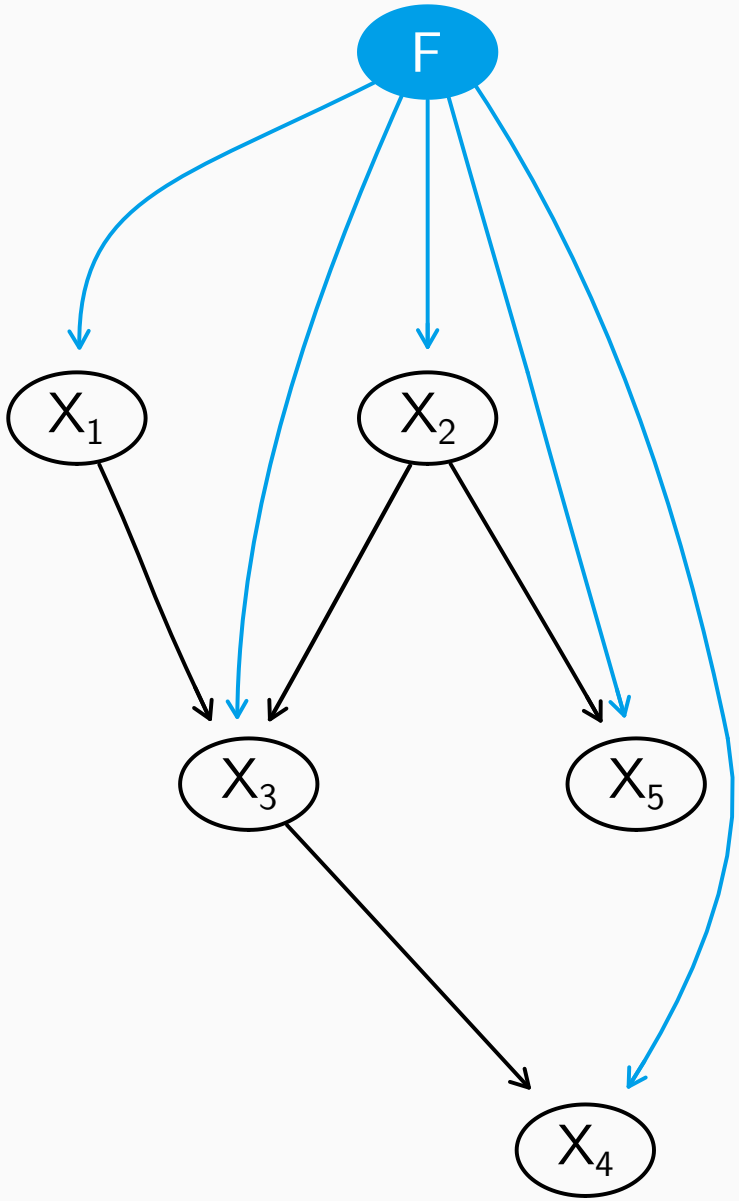
Related Data Sets

The aim: learning the structure of a BN from a set of related data sets identified by F, which is assumed known.

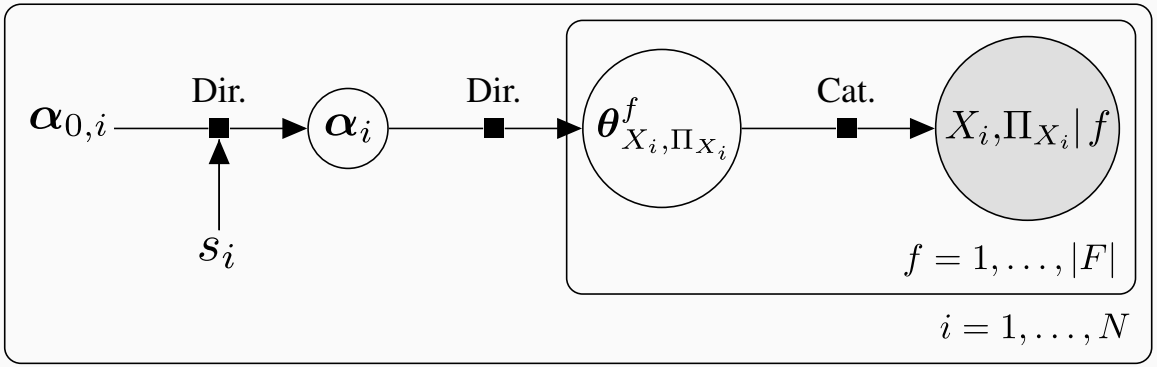
The approach: we would like to do that by pooling information across different data sets to distil structural features that are common to all of them.

The mathematical formulation:

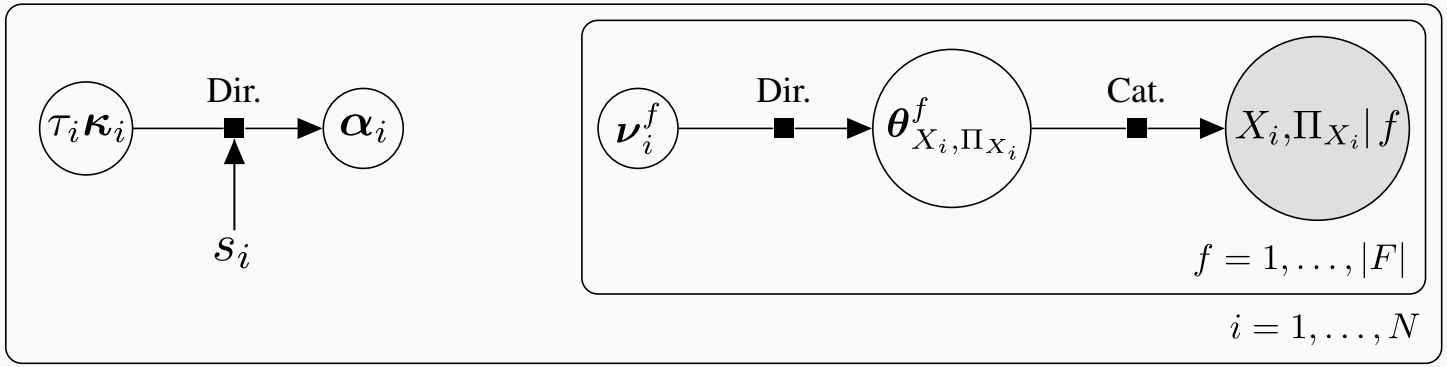
- for discrete variables, a variational Bayesian Dirichlet score with a hierarchical prior (BHD) [1];
- for continuous variables, using mixed-effects models [10].



The Hierarchical Model Behind BHD



Hierarchical Model



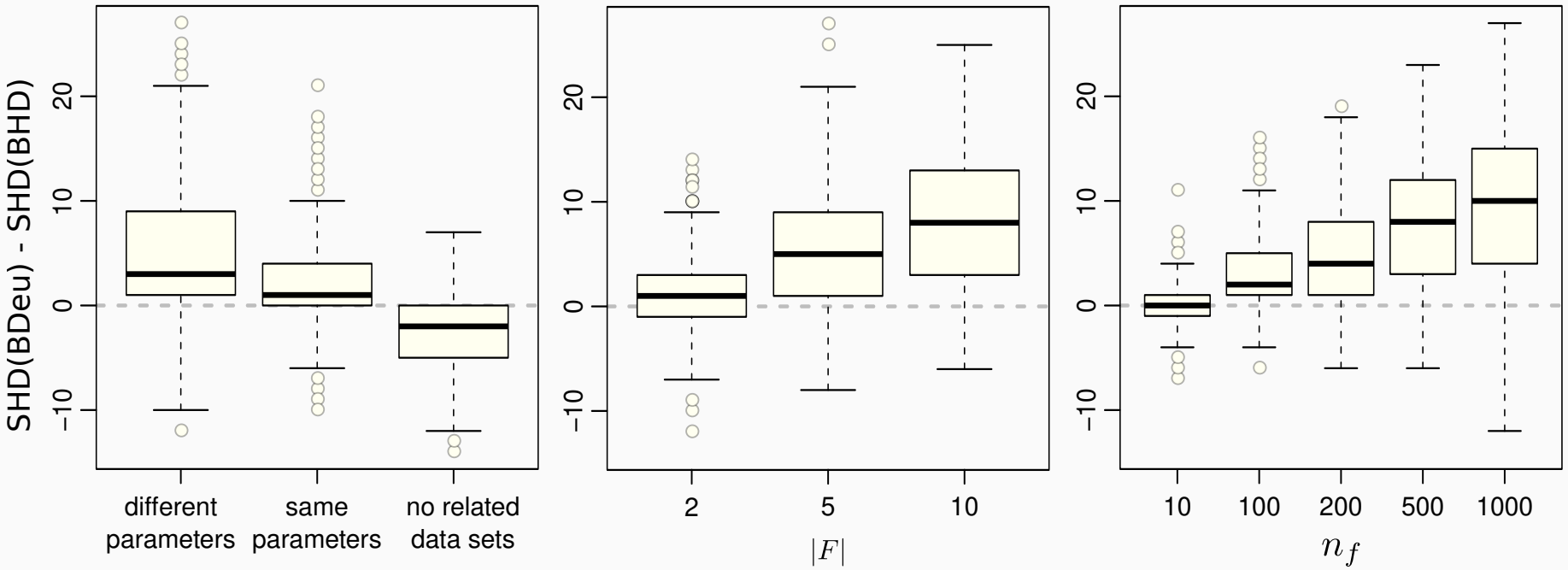
Variational Approximation

Thus we get BHD:

$$\mathrm{P}(\mathcal{D} \mid F, \mathcal{G}) \approx \prod_{i=1}^{N} \prod_{f=1}^{|F|} \prod_{j=1}^{|\Pi_{X_i}|} \left[\frac{\Gamma(s_i \hat{\kappa}_{ij})}{\Gamma(s_i \hat{\kappa}_{ij} + n^f_{ij})} \prod_{k=1}^{|X_i|} \frac{\Gamma(s_i \hat{\kappa}_{ijk} + n^f_{ijk})}{\Gamma(s_i \hat{\kappa}_{ijk})} \right]$$

where $s_i \hat{\kappa}_{ijk}$ = the posterior mean of α_{ijk} under the variational model.

BHD Versus BDeu



The BHD score:

- has better structural accuracy than BDeu when we are modelling related data sets;
- it gets increasingly better as the number of related grows;
- it gets increasingly better as the size of (at least some of) the individual related data sets grows.

What About Continuous Variables?

In a Gaussian BN, each node X_i has distribution

$$X_i = \mu_{X_i} + \Pi_{X_i}\beta_{X_i} + \varepsilon_{X_i} \qquad \text{with} \qquad \varepsilon_{X_i} \sim N(0, \sigma^2_{X_i}\mathbf{I}_n). \quad (1)$$

Adding the node F would make it a conditional Gaussian BN in which we fit a separate linear regression for each data set j identified by F:

$$X_i = \mu_{ij} + \Pi_{X_i}\beta_{ij} + \varepsilon_{X_i} \qquad \text{with} \qquad \varepsilon_{X_i} \sim N(0, \sigma^2_{ij}\mathbf{I}_{n_j}). \quad (2)$$

A mixed-effects model that takes (1) and adds random effects for all Π_{X_i}

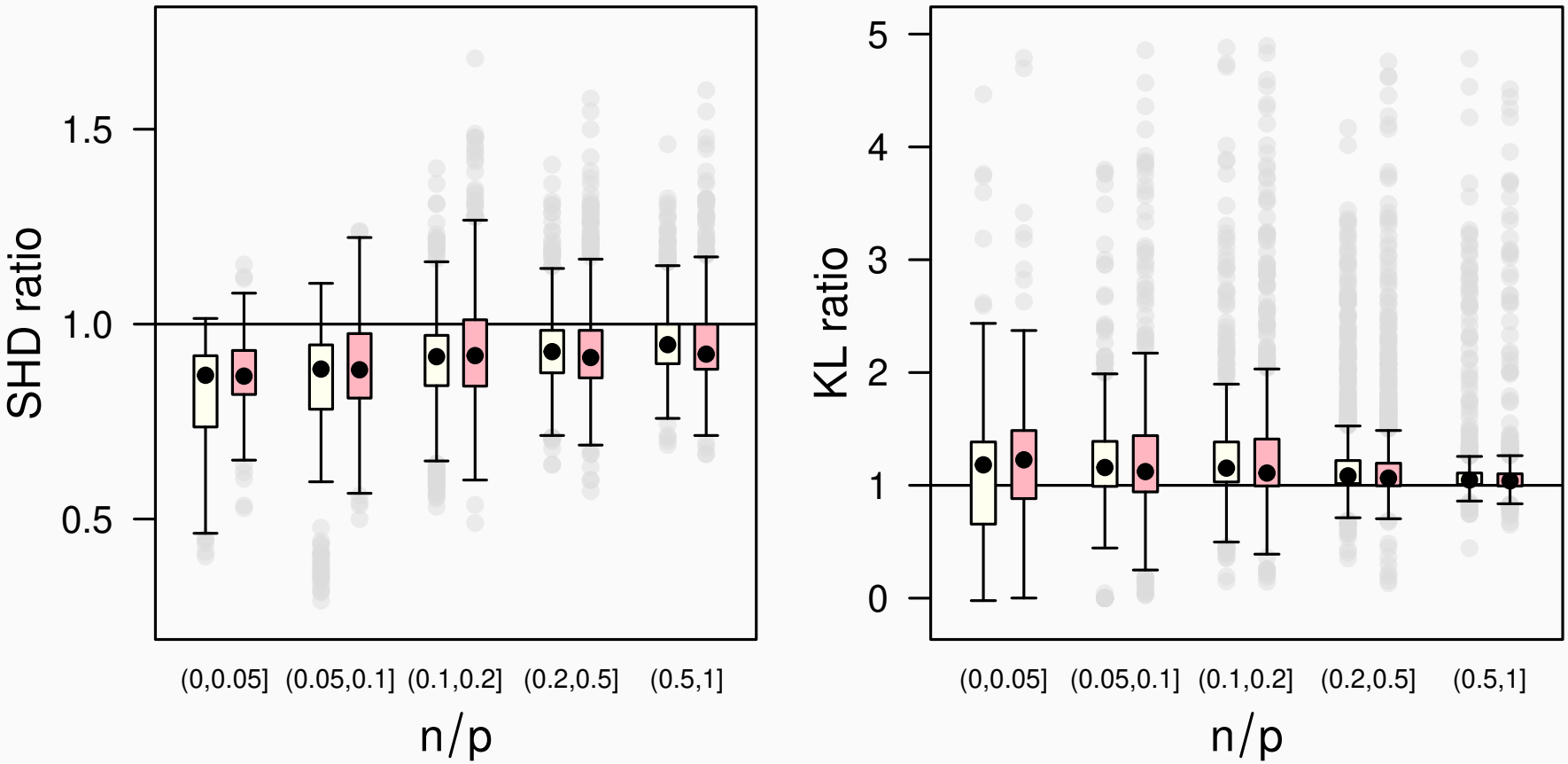
$$X_i = \mu_{X_i} + \Pi_{X_i}\beta_{X_i} + \mathbf{Z}\mathbf{b}_{X_i} + \varepsilon_{X_i},\ \mathbf{b}_{X_i} \sim N(\mathbf{0}, \Sigma), \varepsilon_{X_i} \sim N(0, \sigma^2_{X_i}\mathbf{I}_n)$$

has the same form as (2),

$$X_i = (\mu_{ij} + b_{0j}) + \Pi_{X_i}(\beta_{X_i} + \mathbf{b}_{ij}) + \varepsilon_{X_i},$$

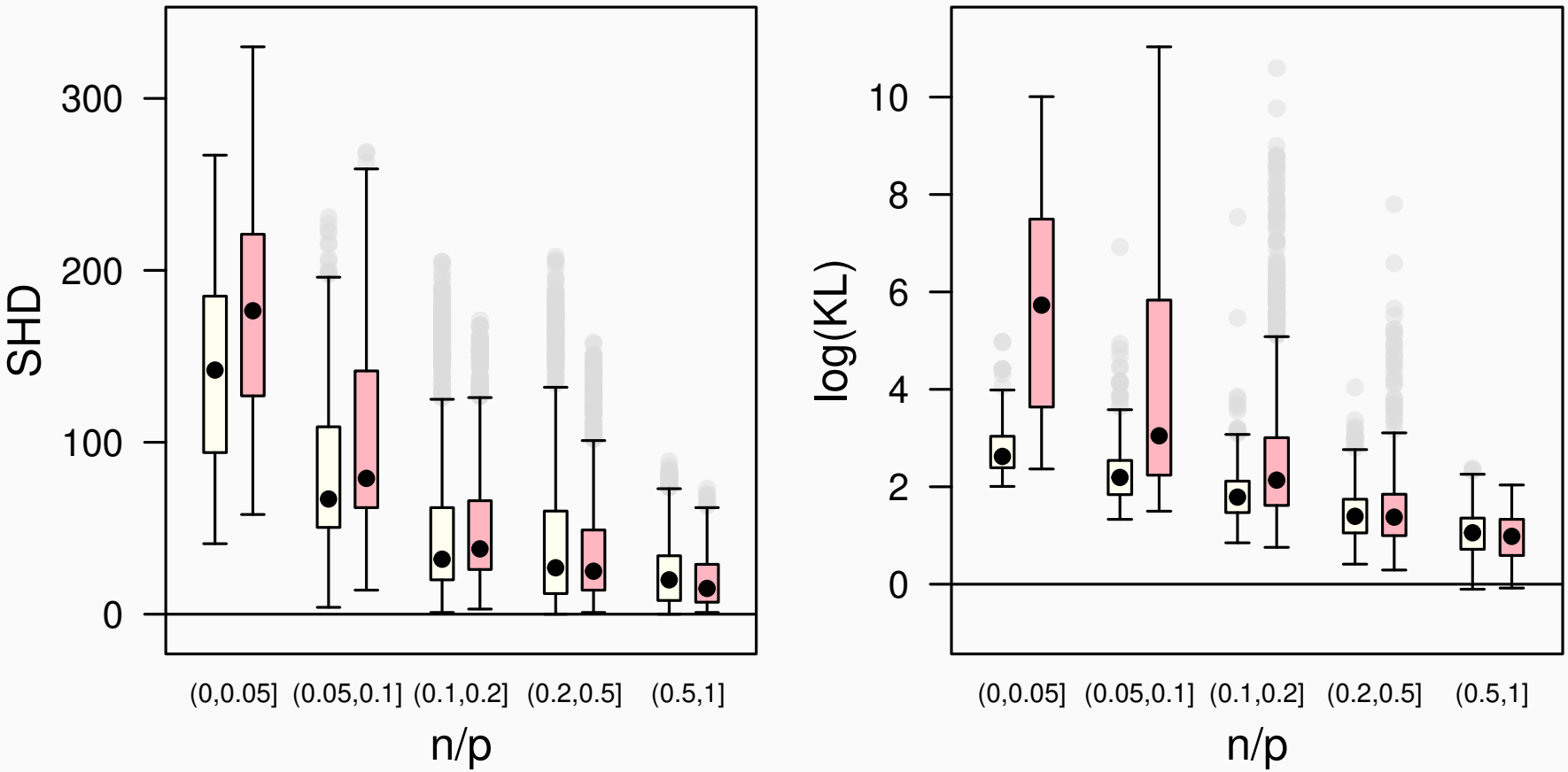
but pools information across data sets much like BHD does [12].

Pooling Versus No Pooling: Homogeneous Data



If the data are just a single homogeneous data set, introducing mixed effects does not degrade performance.

Pooling Versus No Pooling: Heterogeneous Data



If the data really are a collation of related data sets, introducing mixed effects improves both structural (SHD) and parametric accuracy (KL). The difference becomes more marked if the related data sets are unbalanced.

We Can do More!

- We can drop the assumption that F is a parent of all other nodes: as long as we have a score that can compare models with and without random effects, we are good.
- We need not to restrict ourselves to Gaussian variables: we can use generalised mixed-effects models as local distributions to handle a diverse set of distributions.
- We can use random effects to model more complex structures in the data:
  - cryptic relatedness (in genetics);
  - spatial dependencies;
  - temporal dependencies.

- ✓ Bayesian Networks: Definition and Assumptions
- ✓ Continuous-Time Bayesian Networks
- ✓ Bayesian Networks for Structured Data
- → **Future Directions**

Conclusions and Future Directions

Bayesian networks are a fundamental tool in machine learning: they subsume many models [11] and handle incomplete data [3], continuous-time time series [4] and collections of related data sets [1].

What next?

- Making CTBNs into Markov decision processes [7, 13] to model as streaming health data where we administer medical treatments in real time.
- A comprehensive approach to related data sets that can handle conditional Gaussian BNs, and thus discrete and Gaussian BNs as particular cases.
- A reanalysis of a complex environmental data set such as [15] to explore BNs with a spatio-temporal structure.

Acknowledgements

Christopher Marquis
École Polytechnique Fédérale de Lausanne (EPFL)

Alessandro Bregoli
Fabio Stella
Università degli Studi di Milano-Bicocca

Søren Wengel Mogensen
Lunds Universitet

Laura Azzimonti
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)

Thanks!

Any questions?

References I

- L. Azzimonti, G. Corani, and M. Scutari.
  A Bayesian Hierarchical Score for Structure Learning from Related Data Sets.
  International Journal of Approximate Reasoning, 142:248–265, 2021.
- L. Bain and M. Englehardt.
  Statistical Analysis of Reliability and Life-Testing Models: Theory and Methods.
  CRC Press, 1991.
- T. Bodewes and M. Scutari.
  Learning Bayesian Networks from Incomplete Data with the Node-Averaged Likelihood.
  International Journal of Approximate Reasoning, 138:145–160, 2021.
- A. Bregoli, M. Scutari, and F. Stella.
  A Constraint-Based Algorithm for the Structural Learning of Continuous-Time Bayesian Networks.
  International Journal of Approximate Reasoning, 138:105–122, 2021.
- D. Colombo and M. H. Maathuis.
  Order-Independent Constraint-Based Causal Structure Learning.
  Journal of Machine Learning Research, 15:3921–3962, 2014.
- D. Heckerman and D. Geiger.
  Learning Bayesian Networks: a Unification for Discrete and Gaussian Domains.
  In *UAI*, pages 274–284, 1995.

References II

- K. F. Kan and C. R. Shelton.
  Solving Structured Continuous-Time Markov Decision Processes.
  In *ISAIM*, 2008.
- B. Mitchell.
  A Comparison of Chi-Square and Kolmogorov-Smirnov Tests.
  The Royal Geographical Society, 3(4):237–241, 1971.
- U. D. Nodelman.
  Continuous Time Bayesian Networks.
  PhD thesis, Stanford University, 2007.
- J. C. Pinheiro and D. M. Bates.
  Mixed-effects models in S and S-PLUS.
  Springer, 2000.
- M. Scutari.
  Bayesian Network Models for Incomplete and Dynamic Data.
  Statistica Neerlandica, 74(3):397–419, 2020.
- M. Scutari, C. Marquis, and L. Azzimonti.
  Using Mixed-Effect Models to Learn Bayesian Networks from Related Data Sets.
  Proceedings of Machine Learning Research (PGM 2022), 2022.

References III

- L. Sturlaugson, L.Perreault, and J. W. Sheppard.
  Factored Performance Functions and Decision Making in Continuous Time Bayesian Networks.
  Journal of Applied Logic, 22:28–45, 2017.
- T. S. Verma and J. Pearl.
  Equivalence and Synthesis of Causal Models.
  In *UAI*, pages 255–268, 1990.
- C. Vitolo, M. Scutari, A. Tucker, and A. Russell.
  Modelling Air Pollution, Climate and Health Data Using Bayesian Networks: a Case Study of the English Regions.
  Earth and Space Science, 5(4):76–88, 2018.