INFERENCE FOR PROBABILISTIC DEPENDENCY GRAPHS

Oliver E Richardson, Joseph Y Halpern, Christopher De Sa

Cornell Bowers C·IS Computer Science

A **PDG** is a directed (hyper) graph, with a conditional probabilities and confidences attached to arcs.

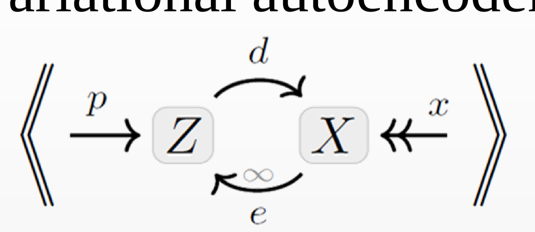
PDGs can capture:

- inconsistent beliefs, and provide a way to measure the degree of this inconsistency;
- Bayesian networks (BNs)

… but PDGs are more modular;

- factor graphs

- variational autoencoders (VAEs)



…. including their standard loss function, as inconsistency

$= -\mathrm{ELBO}_{p,e,d}(x)$.

- other classical learning settings and their loss functions, statistical divergences, regularizers, GANs, learning algorithms, causal models, …

PDG FORMALISM & SEMANTICS

$\mathcal{M} =$ variables $\mathcal{X}$ connected by arcs $\mathcal{A}$; each $(S \xrightarrow{a} T) \in \mathcal{A}$ is associated with:

a conditional probability $\mathbb{P}_a(T|S)$, and two confidences: β_a (observational) and α_a (structural).

A joint probability $\mu(\mathcal{X})$ can be incompatible with a PDG in two ways:

Observational Incompatibility with $(\mathbb{P}, \boldsymbol{\beta})$: $\sum_{S \xrightarrow{a} T \in \mathcal{A}} \beta_a \, D\big(\mu(T,S) \,\|\, \mathbb{P}_a(T|S)\mu(S)\big)$

Structural Incompatibility with $(\mathcal{A}, \boldsymbol{\alpha})$: $\Big(\sum_{S \xrightarrow{a} T \in \mathcal{A}} \alpha_a \, \mathrm{H}_\mu(T|S)\Big) - \mathrm{H}(\mu)$

cvx!

$$[\![\mathcal{M}]\!]_\gamma(\mu) := \mathit{OInc}_{\mathcal{M}}(\mu) + \gamma\, \mathit{SInc}_{\mathcal{M}}(\mu)$$

$$= \mathbb{E}_\mu \Big[\sum_{S \xrightarrow{a} T \in \mathcal{A}} \log \frac{\mu(T|S)^{\beta_a - \gamma\alpha_a}}{\mathbb{P}_a(T|S)^{\beta_a}} \Big] - \gamma\, \mathrm{H}(\mu)$$

overall incompatibility, placing weight $\gamma \geq 0$ on the structural information.

Tasks:

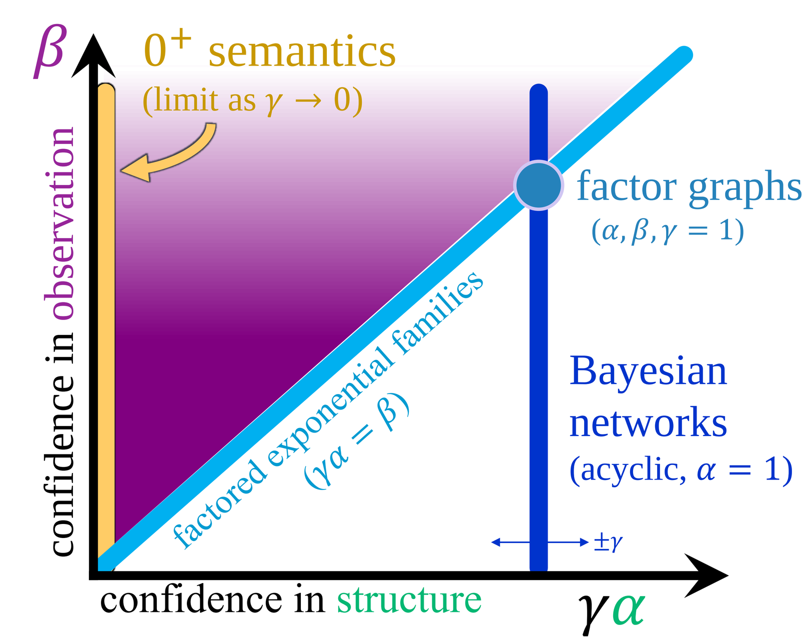
- **γ-inconsistency** $\langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma$**:** find the minimum value of this function.
- **γ-inference:** answer questions about all minimizing distribution(s).
  - └→ **0^+-inference**: the observational limit (behavior as $\gamma \to 0$):
    - focuses on observation, using structure only to break ties;
    - produces a unique distribution.

Q: *How to calculate $Pr(Y|X)$ in a PDG? or its degree of inconsistency?*

A: Often, can translate PDG scoring function to a small convex optimization problem with "exponential cone" constraints, answering both questions.

INFERENCE LANDSCAPE

- Easily converted to a factor graph; inference with belief propagation — $\tilde{O}(N)$
- Inference now possible with exponential conic programming — $\tilde{O}(N^4)$
- (no known inference algorithm; problem may not be convex)



PDG Inference is #P hard, as it subsumes BN inference.

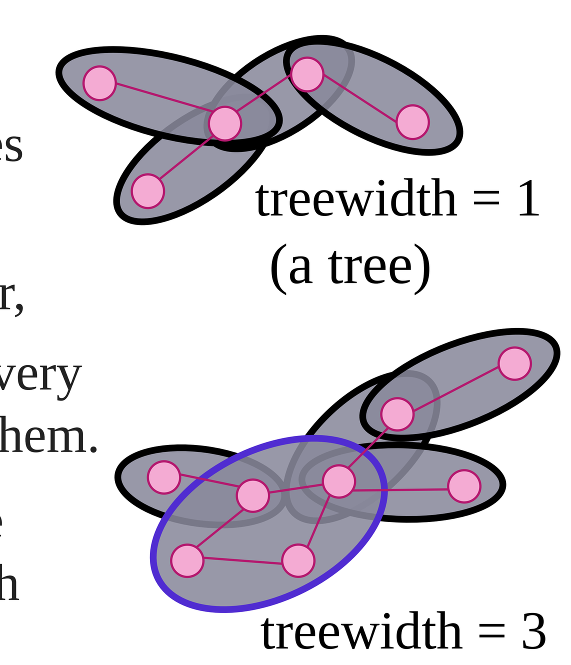
(background)

…but inference in BNs is **efficient for trees**, and graphs G that are sufficiently "tree-like".

Defn. A *tree decomposition* of G is a tree whose nodes are subsets of vertices of G, called *clusters*, such that:

- each (hyper) edge of G is contained in some cluster,
- the intersection of any two clusters is a subset of every cluster on the unique path (since it's a tree) between them.

The *width* of a tree decomposition is one less than the largest cluster; the *treewidth* of G is the smallest width of any tree decomposition of G.



Is inference efficient for tree-like PDGs?

THEOREM: POLYNOMIAL TIME PDG INFERENCE UNDER BOUNDED TREEWIDTH

For $\gamma \in \{0^+\} \cup \Big(0, \min_{a \in \mathcal{A}} \frac{\beta_a}{\alpha_a}\Big]$, can do γ-inference and calculate γ-inconsistency

for a PDG that has N total arcs + vars, treewidth T, and gap $\beta^{\max} / \beta^{\min}$ between the largest and smallest confidences,

to precision ϵ, in time

$$O\Big(N^4 \Big(T + \log \frac{N}{\epsilon} \frac{\beta^{\max}}{\beta^{\min}}\Big) 2^T\Big) \subseteq \tilde{O}(N^4).$$

(if T is bounded)

THEOREM: calculating inconsistency is closely related to inference & just as difficult.

a) Calculating the degree of inconsistency is #P-hard;

b) There's a linear reduction from γ-inference to the problem of calculating γ-inconsistency.

KEY IDEAS

1. Insight: can rewrite the PDG scoring function (when convex) as an *exponential conic program*: an optimization problem of the form

$$\underset{\mathbf{x}}{\text{minimize}} \;\; \mathbf{c}^{\mathsf{T}}\mathbf{x} \quad \text{subject to} \;\; A\mathbf{x} = \mathbf{b}, \;\; \mathbf{x} \in K$$

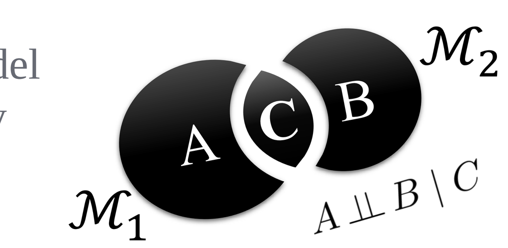
K is a product of positive orthants ($x \geq 0$), and "exponential cones", which are related to relative entropy.

Such problems can be solved by in time $O(\mathrm{poly}(\dim K))$.

2. For **0^+-inference** (limit as $\gamma \to 0$), need to minimize *SInc* among minimizers of *OInc*. This set of *OInc*-minimizers is characterized by shared marginals, so can use linear constraints after finding one minimizer of *OInc*. Then, *SInc* becomes convex!

3. Intractable to optimize $[\![\mathcal{M}]\!]_\gamma(\mu)$ over joint distributions μ, which grow exponentially in # of vars. A *clique tree* is a probability over every cluster of a tree decomposition of $\mathcal{M}$, and represents μ satisfying an important independence property of PDGs:

Theorem (PDG *Markov Property*). In the combined model $\mathcal{M}_1 + \mathcal{M}_2$, the variables of $\mathcal{M}_1$ and $\mathcal{M}_2$ are conditionally independent given the ones they have in common.

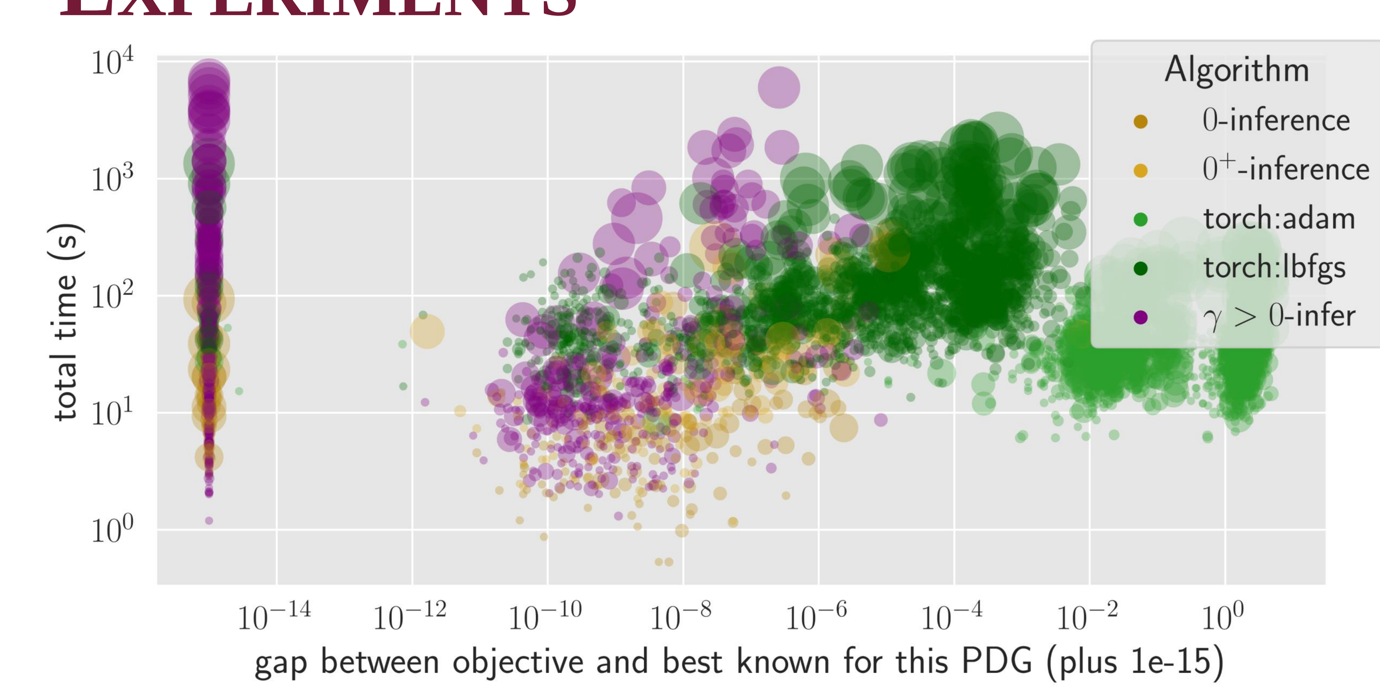


Thus, it suffices to optimize over clique trees, which grow linearly in # vars, given our assumption of bounded treewidth. Formulating the optimization over clique trees involves some additional subtleties (because of *SInc*)… see the paper for details!

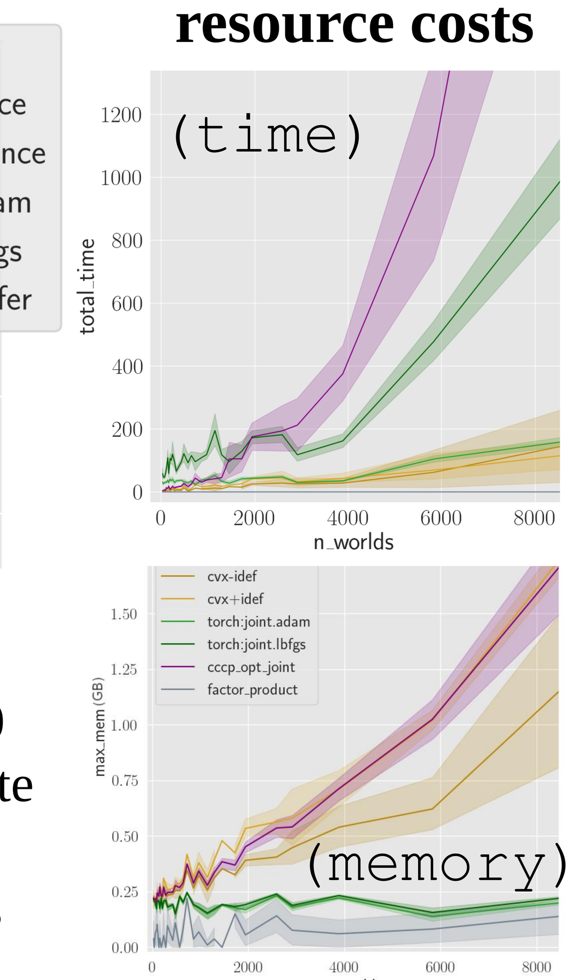
IMPLEMENTATION

connects PDG python library to commercial solvers such as `MOSEK` and `ECOS`, via `cvxpy`.

EXPERIMENTS

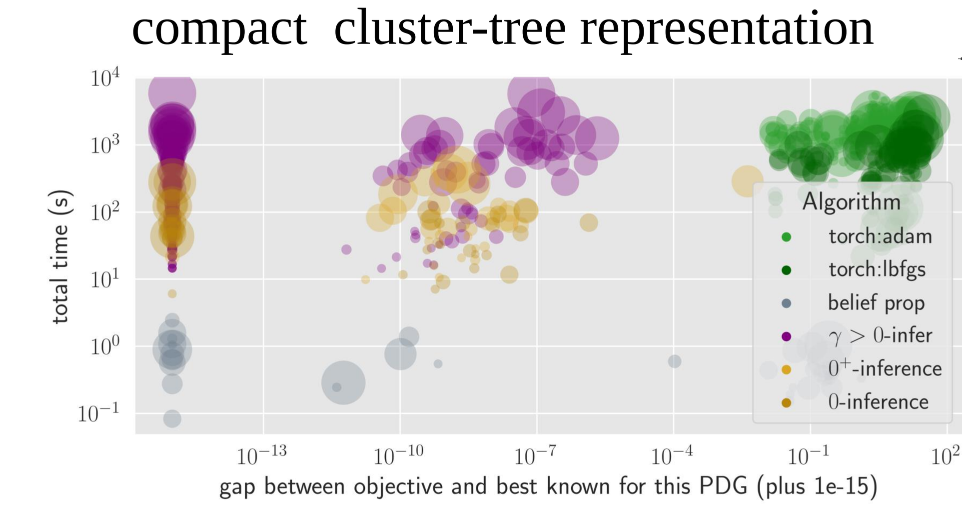


resource costs



Scatter plot: running inference on random PDGs with ~10 variables. The convex solver (gold, violet) is more accurate (←) than black-box optimization baselines (greens), and often faster (↓) for small PDGs. The area of each circle is proportional to the size of the optimization problem.

A similar synthetic experiment, this time with random k-trees, using the compact cluster-tree representation



Results on a BN dataset. (But in this case, we can use belief propagation, which is strictly better).