THE LOCAL INCONSISTENCY RESOLUTION ALGORITHM

Oliver E Richardson

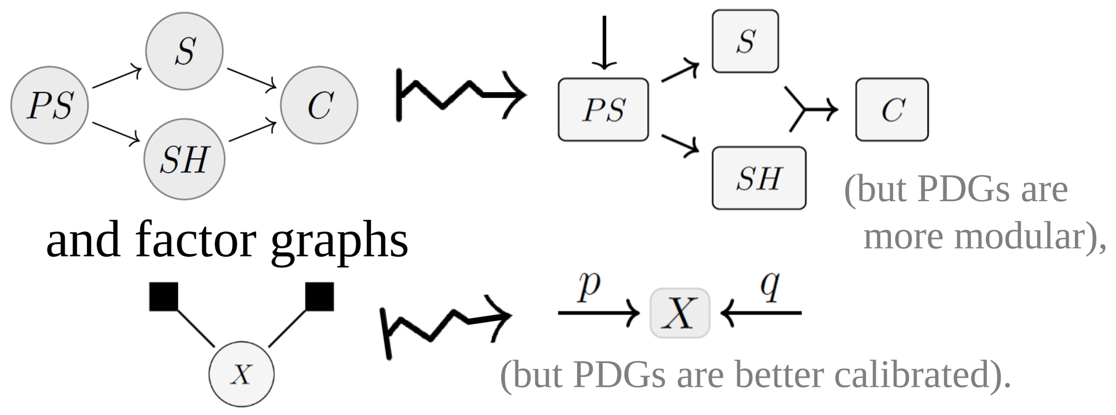
Cornell Bowers CIS Computer Science

A generic algorithm for learning and (approximate) inference, with an intuitive epistemic interpretation. Unifies many important algorithms.

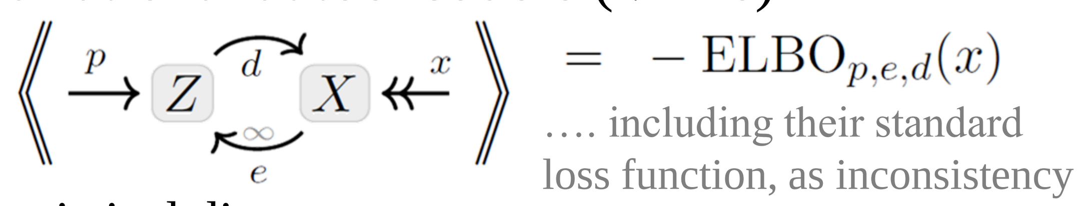
Key Representation: Probabilistic Dependency Graphs (PDGs) are directed (hyper) graphs with probabilities and confidences attached to edges.

PDGs can capture:

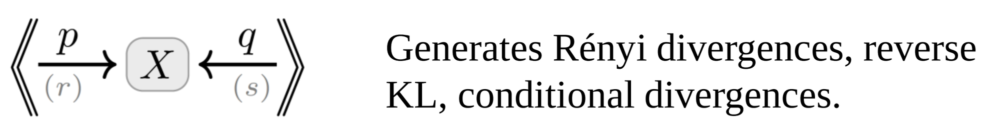
- inconsistent beliefs, providing a natural way to measure the degree of this inconsistency;
- **graphical models**, such as Bayesian networks



- **learning settings** and their **loss functions**, e.g.,
  - variational autoencoders (VAEs)



  - statistical divergences



  - regularizers as priors, accuracy, MSE, ….

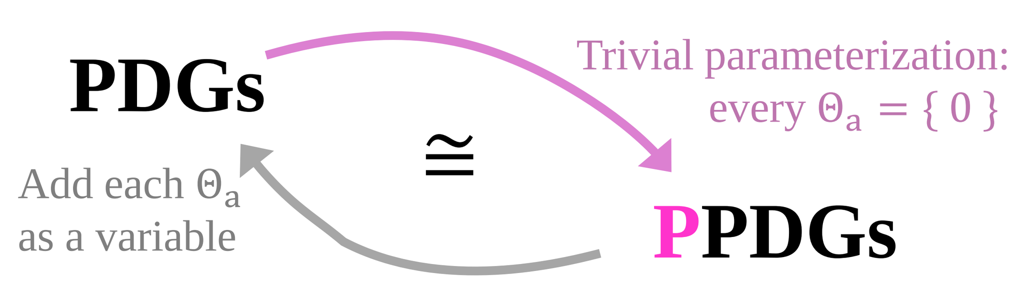
FORMALISM: PARAMETERIZED PDGS

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

- a convex parameter space $\Theta_a \subseteq \mathbb{R}^n$
- a conditional probability $\mathbb{P}_a(T|S,\Theta_a)$,
- two confidences: β_a (observational) and α_a (structural).

Fix a parameter setting $\theta \in \prod_{a\in\mathcal{A}} \Theta_a$, to get an (ordinary) PDG $\mathcal{M}(\theta)$.

The two variants are equivalent.



Inconsistency semantics.

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

Observational Incompatibility with $(\mathbb{P}, \boldsymbol{\beta})$

$$\sum_{S \xrightarrow{a} T \in \mathcal{A}} \beta_a \, \boldsymbol{D}\Big(\mu(T,S) \,\Big\|\, \mathbb{P}_a(T|S)\mu(S)\Big)$$

Structural Deficiency with $(\mathcal{A}, \boldsymbol{\alpha})$

$$\mathbb{E}_{\mu}\left[\log \frac{\mu(\mathcal{X})}{\lambda(\mathcal{X})} \prod_{S \xrightarrow{a} T} \left(\frac{\lambda(T|S)}{\mu(T|S)}\right)^{\alpha_a}\right]$$

Degree of inconsistency

$$\langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma := \inf_{\mu} \Big(OInc_{\mathcal{M}}(\mu) + \gamma \, SDef_{\mathcal{M}}(\mu) \Big)$$

placing weight $\gamma \geq 0$ on the structural information

is the smallest possible incompatibility with any $\mu(\mathcal{X})$.

Algorithm: Local Inconsistency Resolution (LIR)

Input: context PDG $\mathcal{C}tx$, mutable memory $\mathcal{M}(\Theta)$.

Initialze $\theta^{(0)}$;

for $t = 0, 1, 2, \ldots$ **do**

$\mathcal{C}tx \leftarrow \textsc{Refresh}(\mathcal{C}tx)$;

1. $\varphi, \chi, \gamma \leftarrow \textsc{Refocus}()$; — select attn and ctrl masks

Write $\exp_\theta(t\,\mathrm{X})$ for the path following vector field X for time t, starting at θ.

Gradient Flow of $f: \Theta \to \mathbb{R}$ starting at θ: $t \mapsto \exp_\theta(t \nabla_\Theta f(\Theta))$

Calculate the inconsistency of the combined context and memory, weighted by attention.

2. $$\theta^{(t+1)} \leftarrow \exp_{\theta^{(t)}}\Big\{ -\chi \odot \nabla_\Theta \big\langle\!\big\langle \varphi \odot \big(\mathcal{C}tx + \mathcal{M}(\Theta)\big) \big\rangle\!\big\rangle_\gamma \Big\};$$

Reduce this inconsistency by (an approximation to) gradient flow, starting at previous state $\theta^{(t)}$, changing each parameter in proportion to our control of it.

LIR IN THE CLASSIFICATION SETTING

Consider a discriminator $p_\theta(Y|X)$ and sample (x, y). Together, they have inconsistency

$$\left\langle\!\!\left\langle \xrightarrow{x}\!\!\!\to \boxed{X} \xrightarrow{p_\theta} \boxed{Y} \leftarrow\!\!\!\xleftarrow{y} \right\rangle\!\!\right\rangle = \log \frac{1}{p_\theta(y|x)}.$$

Can resolve by modifying:

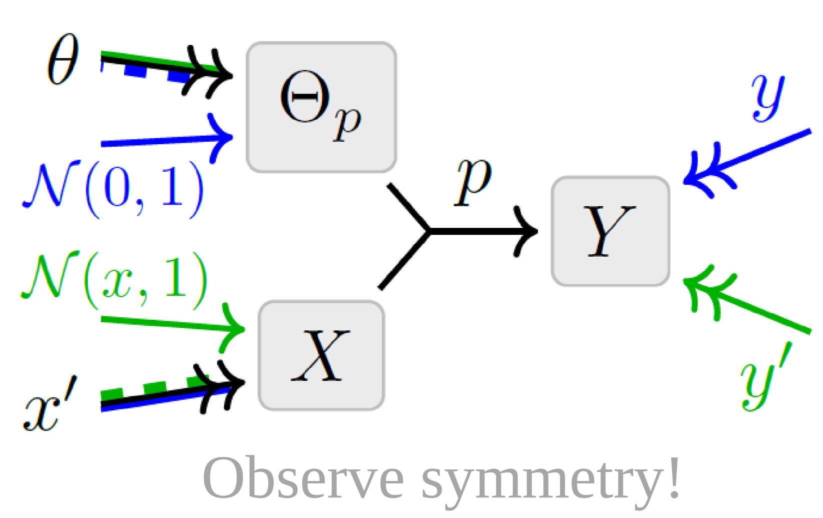
- θ, to train the discriminator
- y, resulting in a forward pass
- x, to form an adversarial example

SGD. Control over p_θ. Replace (x,y) with empirical distribution over a batch, and suppose $\textsc{Refresh}(\mathcal{C}tx)$gets a new batch. This performs SGD with learning rate $\chi(p) \cdot \varphi(p)$.

Adversarial Training.

Add discriminator params as a variable Θ_p with Gaussian prior.

- Construct attack $x' \approx x$ that p misclassifies as y'
- Patch p to classify x' as y

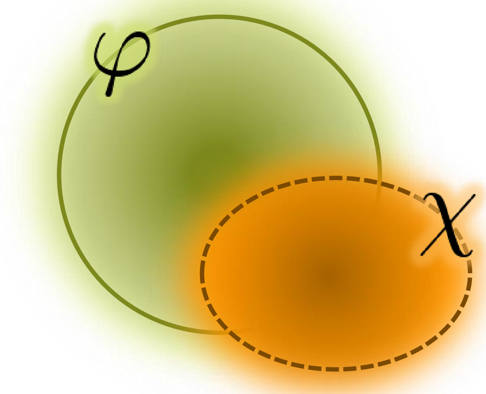


What causes changes in beliefs? Some say it is internal conflict. But identifying inconsistencies is difficult. So in practice, we resolve them *locally:* looking only at a small part of the picture, and changing only another small part at a time.

FOCUS: ATTENTION AND CONTROL

attend only to probabilities of a subset of arcs $A \subseteq \mathcal{A}$ (or attn mask φ)

control only parameters of a subset of arcs $C \subseteq \mathcal{A}$ (or ctrl mask χ)

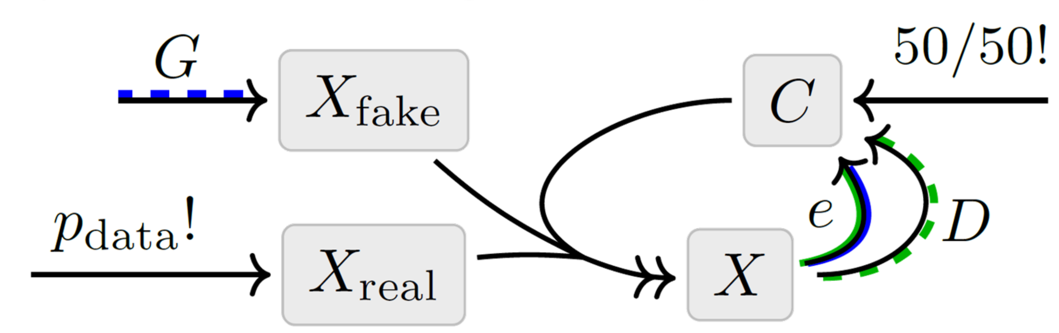


Typical use case: select focus (φ, χ) from a fixed set of *foci* $\mathbf{F} = \{$ ■, ■, …$\}$.

MORE EXAMPLES

- Training Generative Adversarial Networks (GANs).

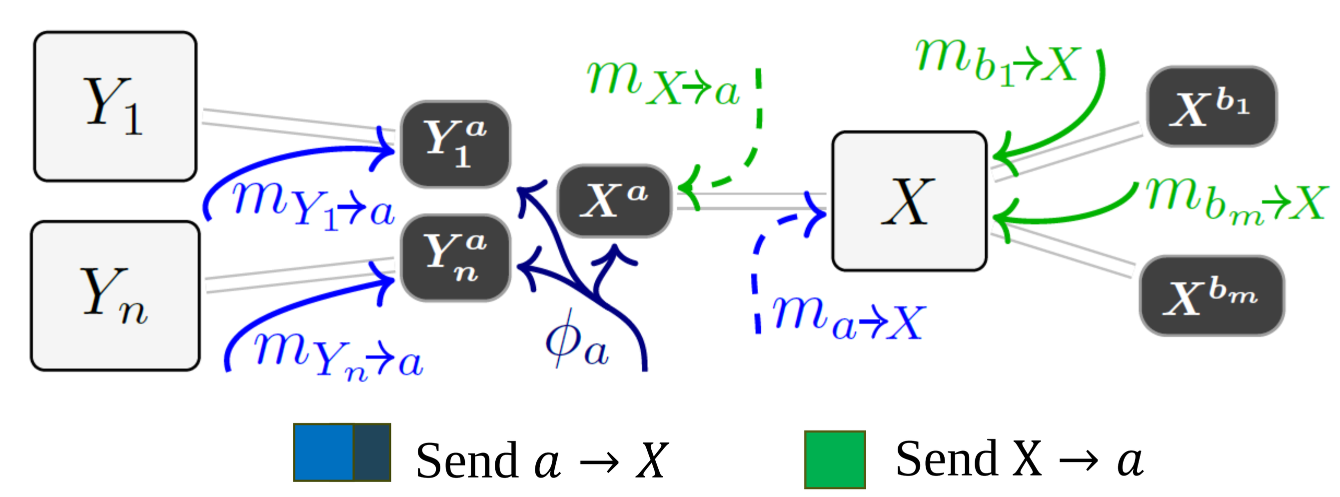
Typically trained with minimax game: $\min_G \max_D \mathcal{L}^{GAN}$



■ *Generator's focus*
inconsistency = JSD(G, p_{data})
or $+\mathcal{L}^{GAN}$ if disbelieves D

■ *Discriminator's focus*
inconsistency = KL(D, D^{opt})
or - $\mathcal{L}^{GAN}$ if disbelieves e

- Message Passing: Sum-Product Belief Propagation



Observation: the message passing equations are sums of products of factors, i.e., do inference in *local* factor graphs.

Here, $\mathcal{M}(\theta)$: collection of messages (BP data structure)

$\mathcal{C}tx$: original factor graph, as a PDG

- Variational Inference, EM algorithm, e.g., VAE training.