Probabilistic Dependency Graphs and Inconsistency

How to model, measure, and mitigate internal conflict

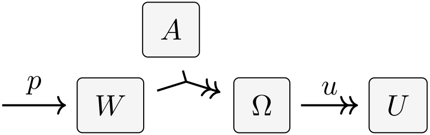
Oliver Richardson

Cornell University
Department of Computer Science

September 2021

The standard way of modeling an agent with uncertainty:

- a probability distribution $p : \Delta W$ over worlds W,
- a utility function $u : \Omega \to \mathbb{R}$, some actions A.



Such agents cannot have internal conflict;

by construction, they have consistent beliefs and desires.

I like this slide

⟨ INCOMPLETE ⟩

Why build systems that can be inconsistent, if inconsistency is bad?

Why entertain the possibility of being wrong, if being wrong is bad?
Elsewhere, computer scientists take great care to model inconsistency:

- assertions and test cases: Text
-
- losses for training neural networks `(totally separate from the probabilistic model)`

I think you're agonizing too much over this. Why not just say "Because sometimes do have inconsistent beliefs. We also want to model the dynamics of making them consistent, and capture the extent to which they're inconsistent.

Yet Another Probabilistic Graphical Model

I don't think that I would remove this but modify it to say something like: not only do we want to model inconsistency, but we want to do so using a probabilistic graphical model.
Then go back to the current slides and say "In doing so, we get much more."

$\langle$ `remove slide, once intro is finished` $\rangle$

We introduce *probabilistic dependency graphs* (PDGs), a new class of graphical models for representing uncertainty.

Yet Another Probabilistic Graphical Model

⟨ `remove slide, once intro is finished` ⟩

We introduce *probabilistic dependency graphs* (PDGs), a new class of graphical models for representing uncertainty.

Why do we need another one?

Yet Another Probabilistic Graphical Model

$\langle$ `remove slide, once intro is finished` $\rangle$

We introduce *probabilistic dependency graphs* (PDGs), a new class of graphical models for representing uncertainty.

Why do we need another one?

- To resolve inconsistency, we must first model it.

Yet Another Probabilistic Graphical Model

⟨ `remove slide, once intro is finished` ⟩

We introduce *probabilistic dependency graphs* (PDGs), a new class of graphical models for representing uncertainty.

Why do we need another one?

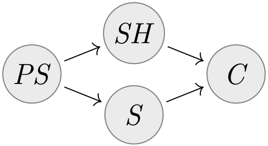
- To resolve inconsistency, we must first model it.
- In doing so, we get much more …

Two aspects of Bayesian Networks (BNs)

Qualitative BN, $\mathcal{G}$

an independence relation on variables

- $X \perp\!\!\!\perp_{\mathcal{G}} Y \mid \mathbf{Pa}(X)$, for all non-descendents Y of X



Two aspects of Bayesian Networks (BNs)

Qualitative BN, $\mathcal{G}$

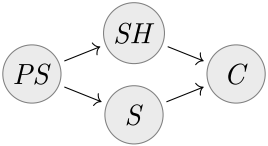
an independence relation on variables

- $X \perp\!\!\!\perp_{\mathcal{G}} Y \mid \mathbf{Pa}(X)$, for all non-descendents Y of X

(Quantitative) BN, $\mathcal{B} = (\mathcal{G}, \mathbf{p})$

a qualitative BN ($\mathcal{G}$) and a cpd $p_X(X \mid \mathbf{Pa}(X))$ for each variable X.

- Defines a joint distribution $\Pr_{\mathcal{B}}$ with the independencies $\perp\!\!\!\perp_{\mathcal{G}}$.



Two aspects of Bayesian Networks (BNs)

Qualitative BN, $\mathcal{G}$

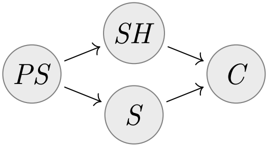
an independence relation on variables

- $X \perp\!\!\!\perp_{\mathcal{G}} Y \mid \mathbf{Pa}(X)$, for all non-descendents Y of X

(Quantitative) BN, $\mathcal{B} = (\mathcal{G}, \mathbf{p})$

a qualitative BN ($\mathcal{G}$) and a cpd $p_X(X \mid \mathbf{Pa}(X))$ for each variable X.

- Defines a joint distribution $\Pr_{\mathcal{B}}$ with the independencies $\perp\!\!\!\perp_{\mathcal{G}}$.



Outline for Section 1

Why is the outline coming after you've introduced BN's? What the high-level story?

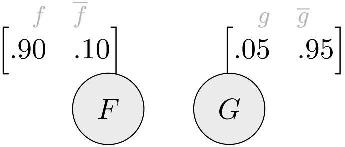
Simple Example: Floomps and Guns

Grok thinks it likely (.95) that guns are illegal,
but that floomps (local slang) are legal (.90).

Simple Example: Floomps and Guns

Grok thinks it likely (.95) that guns are illegal,
but that floomps (local slang) are legal (.90).

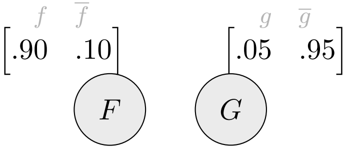
BN



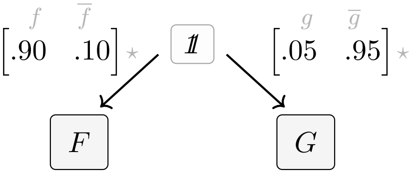
Simple Example: Floomps and Guns

Grok thinks it likely (.95) that guns are illegal,
but that floomps (local slang) are legal (.90).

BN



PDG

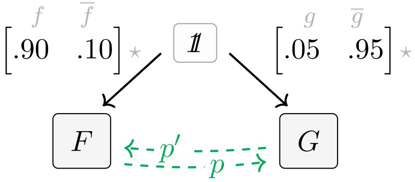


- The cpds of a PDG are attached to edges, not nodes.

Simple Example: Floomps and Guns

BN

PDG

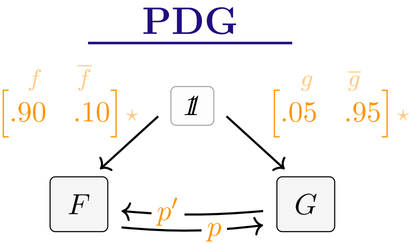


- The cpds of a PDG are attached to edges, not nodes.
- PDGs can incorporate arbitrary new probabilistic information.

Grok learns that Floomps and Guns have the same legal status (92%)

$$p(G \mid F) = \begin{array}{cc} & \begin{array}{cc} g & \bar{g} \end{array} \\ \begin{array}{c} f \\ \bar{f} \end{array} & \begin{bmatrix} .92 & .08 \\ .08 & .92 \end{bmatrix} \end{array} = \big(\, p'(F \mid G) \,\big)^{\mathsf{T}}$$

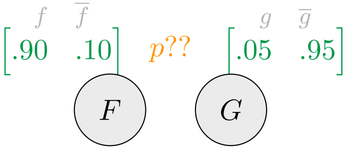
Simple Example: Floomps and Guns



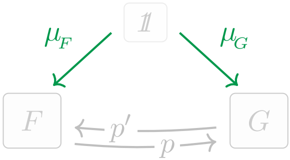
- The cpds of a PDG are attached to edges, not nodes.
- PDGs can incorporate arbitrary new probabilistic information.
- PDGs can be inconsistent

Simple Example: Floomps and Guns

BN



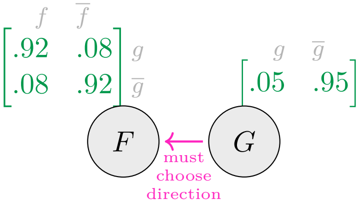
PDG



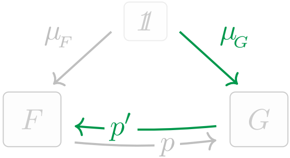
- The cpds of a PDG are attached to edges, not nodes.
- PDGs can incorporate arbitrary new probabilistic information.
- PDGs can be inconsistent,
  - . . . but BNs must resolve inconsistency first,
    which may break symmetry and irrecoverably lose information.

Simple Example: Floomps and Guns

BN



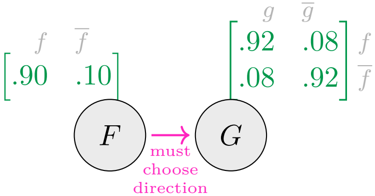
PDG



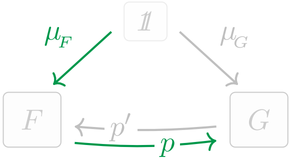
- The cpds of a PDG are attached to edges, not nodes.
- PDGs can incorporate arbitrary new probabilistic information.
- PDGs can be inconsistent,
  - . . . but BNs must resolve inconsistency first,
    which may break symmetry and irrecoverably lose information.

Simple Example: Floomps and Guns

BN

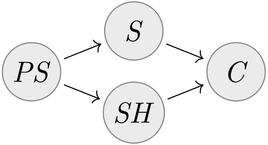


PDG

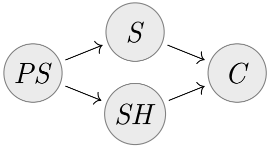


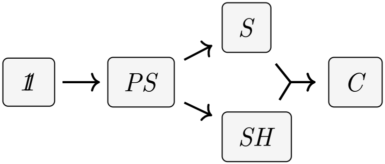
- The cpds of a PDG are attached to edges, not nodes.
- PDGs can incorporate arbitrary new probabilistic information.
- PDGs can be inconsistent,
  - …but BNs must resolve inconsistency first,
    which may break symmetry and irrecoverably lose information.

Bayesian Networks as PDGs

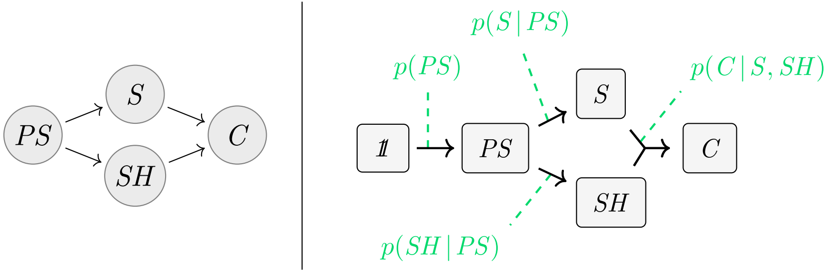


Bayesian Networks as PDGs





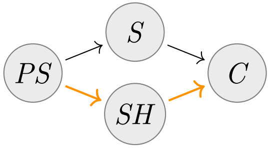
Bayesian Networks as PDGs

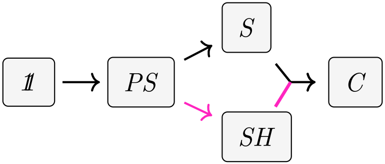


In contrast with BNs:

- edge composition has *quantitative* meaning, since edges have cpds;

Bayesian Networks as PDGs

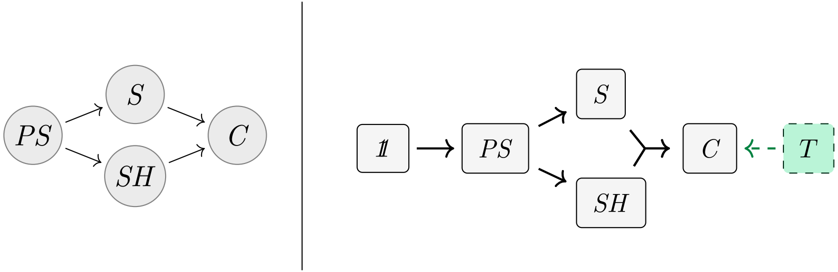




In contrast with BNs:

- edge composition has *quantitative* meaning, since edges have cpds;

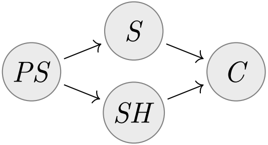
Bayesian Networks as PDGs

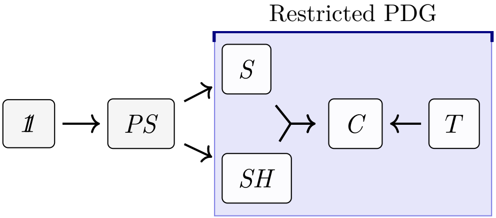


In contrast with BNs:

- edge composition has *quantitative* meaning, since edges have cpds;
- a variable can be the target of more than one cpd;

Bayesian Networks as PDGs

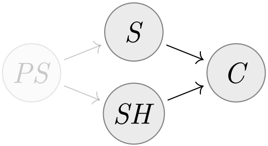




In contrast with BNs:

- edge composition has *quantitative* meaning, since edges have cpds;
- a variable can be the target of more than one cpd;
- arbitrary restrictions of PDGs are still PDGs.

Bayesian Networks as PDGs

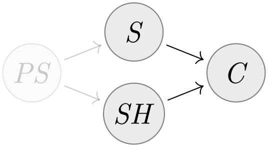


Must now give distributions on SH and S, or distinguish them as "observed" (a *conditional* BN).

In contrast with BNs:

- edge composition has *quantitative* meaning, since edges have cpds;
- a variable can be the target of more than one cpd;
- arbitrary restrictions of PDGs are still PDGs.
  - The analogue is false for BNs!

Bayesian Networks as PDGs



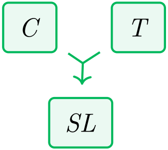
Must now give distributions on SH and S, or distinguish them as "observed" (a *conditional* BN).

In a qualitative BN: *removing* data results in *new* knowledge: $A \perp\!\!\!\perp C$.

In contrast with BNs:

- edge composition has *quantitative* meaning, since edges have cpds;
- a variable can be the target of more than one cpd;
- arbitrary restrictions of PDGs are still PDGs.
  - The analogue is false for BNs!

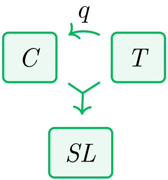
Combining PDGs



Grok wants to be supreme leader (SL).

- She notices that those who use tanning beds have more power, unless they get cancer

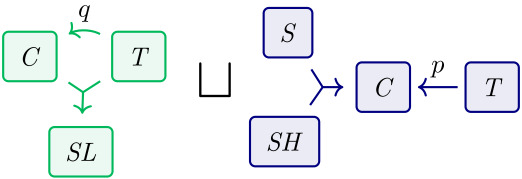
Combining PDGs



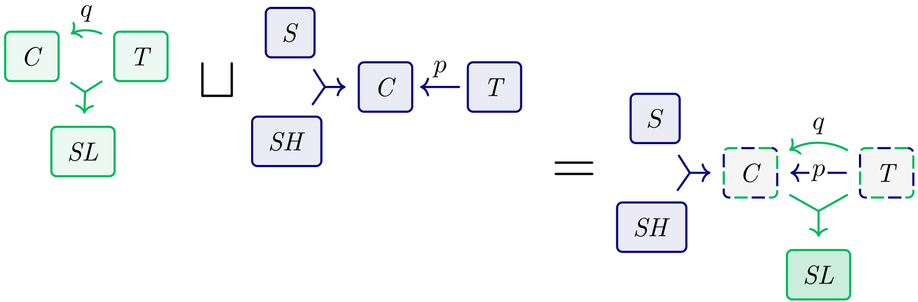
Grok wants to be supreme leader (SL).

- She notices that those who use tanning beds have more power, unless they get cancer
- ...but mom says $q(C \mid T) = \begin{array}{cc} \begin{matrix} c & \bar{c} \end{matrix} & \\ \begin{bmatrix} .15 & .85 \\ .02 & .98 \end{bmatrix} & \begin{matrix} t \\ \bar{t} \end{matrix} \end{array}.$

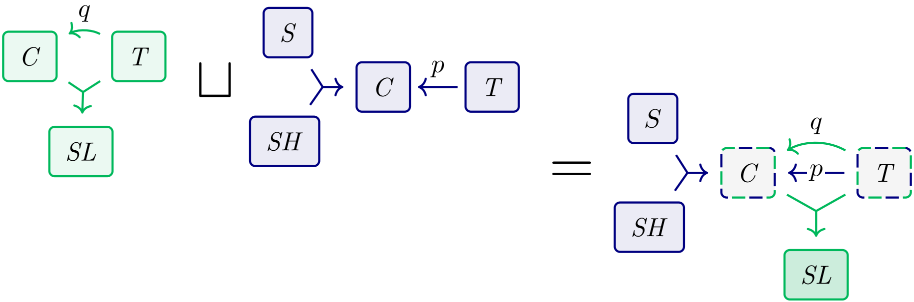
Combining PDGs



Combining PDGs

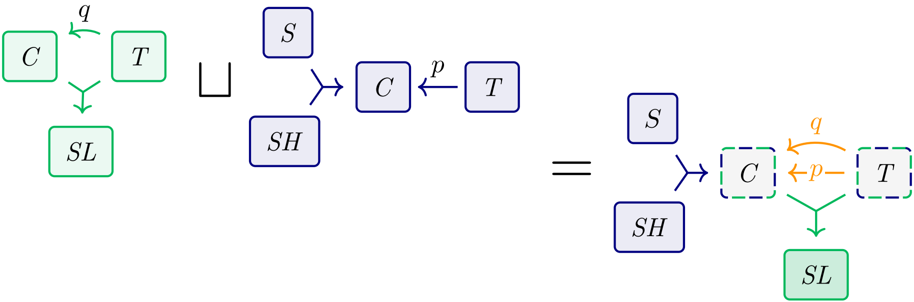


Combining PDGs



- Arbitrary PDGs may be combined without loss of information

Combining PDGs



- Arbitrary PDGs may be combined without loss of information
- They may have parallel edges which directly conflict.

Outline for Section 2

Definition (Probabilistic Dependency Graph)

A PDG is a tuple $\mathcal{m} = (\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p}, \alpha, \beta)$,

Definition (Probabilistic Dependency Graph)

A PDG is a tuple $\boldsymbol{m} = (\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p}, \alpha, \beta)$, where

$\mathcal{N}$ is a finite set of nodes (variables)

Definition (Probabilistic Dependency Graph)

A PDG is a tuple $\mathcal{M} = (\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p}, \alpha, \beta)$, where

$\mathcal{N}$ is a finite set of nodes (variables)

$\mathcal{V}$ gives a set $\mathcal{V}(X)$ of possible values for each X;

Definition (Probabilistic Dependency Graph)

A PDG is a tuple $\mathcal{m} = (\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p}, \alpha, \beta)$, where

$\mathcal{N}$ is a finite set of nodes (variables)

$\mathcal{V}$ gives a set $\mathcal{V}(X)$ of possible values for each X;

$$\mathcal{V}(\mathcal{m}) := \prod_{X \in \mathcal{N}} \mathcal{V}(X)$$

is the set of possible joint variable settings.

Definition (Probabilistic Dependency Graph)

A PDG is a tuple $\boldsymbol{m} = (\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p}, \alpha, \beta)$, where

$\mathcal{N}$ is a finite set of nodes (variables)

$\mathcal{V}$ gives a set $\mathcal{V}(X)$ of possible values for each X;

$\mathcal{E}$ is a set of labeled edges $\{X \xrightarrow{L} Y\}$, (or hyper-edges)

Definition (Probabilistic Dependency Graph)

A PDG is a tuple $\mathcal{m} = (\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p}, \alpha, \beta)$, where

$\mathcal{N}$ is a finite set of nodes (variables)

$\mathcal{V}$ gives a set $\mathcal{V}(X)$ of possible values for each X;

$\mathcal{E}$ is a set of labeled edges $\{X \xrightarrow{L} Y\}$, (or hyper-edges)
and associated to each $X \xrightarrow{L} Y$, there is:

Definition (Probabilistic Dependency Graph)

A PDG is a tuple $\mathcal{m} = (\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p}, \alpha, \beta)$, where

$\mathcal{N}$ is a finite set of nodes (variables)

$\mathcal{V}$ gives a set $\mathcal{V}(X)$ of possible values for each X;

$\mathcal{E}$ is a set of labeled edges $\{X \xrightarrow{L} Y\}$, (or hyper-edges)

and associated to each $X \xrightarrow{L} Y$, there is:

$\mathbf{p}_L$ a cpd $\mathbf{p}_L(Y \mid X)$;

Definition (Probabilistic Dependency Graph)

A PDG is a tuple $\mathcal{M} = (\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p}, \alpha, \beta)$, where

$\mathcal{N}$ is a finite set of nodes (variables)

$\mathcal{V}$ gives a set $\mathcal{V}(X)$ of possible values for each X;

$\mathcal{E}$ is a set of labeled edges $\{X \xrightarrow{L} Y\}$, (or hyper-edges)
and associated to each $X \xrightarrow{L} Y$, there is:

$\mathbf{p}_L$ a cpd $\mathbf{p}_L(Y \mid X)$;

β_L a confidence in the reliability of $\mathbf{p}_L$.

Definition (Probabilistic Dependency Graph)

A PDG is a tuple $\mathcal{m} = (\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p}, \alpha, \beta)$, where

$\mathcal{N}$ is a finite set of nodes (variables)

$\mathcal{V}$ gives a set $\mathcal{V}(X)$ of possible values for each X;

$\mathcal{E}$ is a set of labeled edges $\{X \xrightarrow{L} Y\}$, (or hyper-edges)

and associated to each $X \xrightarrow{L} Y$, there is:

$\mathbf{p}_L$ a cpd $\mathbf{p}_L(Y \mid X)$;

β_L a confidence in the reliability of $\mathbf{p}_L$.

write "p!" for the limit $(\beta_p \to \infty)$ of high confidence in a cpd p.

Definition (Probabilistic Dependency Graph)

A PDG is a tuple $\mathcal{m} = (\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p}, \alpha, \beta)$, where

$\mathcal{N}$ is a finite set of nodes (variables)

$\mathcal{V}$ gives a set $\mathcal{V}(X)$ of possible values for each X;

$\mathcal{E}$ is a set of labeled edges $\{X \xrightarrow{L} Y\}$, (or hyper-edges)

and associated to each $X \xrightarrow{L} Y$, there is:

$\mathbf{p}_L$ a cpd $\mathbf{p}_L(Y \mid X)$;

α_L a confidence in the functional dependence $X \to Y$;

β_L a confidence in the reliability of $\mathbf{p}_L$.

Needless to say, I think that all the following black slides are a *terrible* idea.
This is not where you want to spend your time. It's a major distraction.

Definition (PDG)

$\mathcal{N}$: **Set** (node set)

$\mathcal{V} : \mathcal{N} \to$ **Set** (node values)

$\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N} \times \mathit{Label}$ (edge set)

For $X \xrightarrow{L} Y \in \mathcal{E}$,

$\mathbf{p}_L : \mathcal{V}(X) \to \Delta\mathcal{V}(Y)$ (edge cpd)

$\alpha_L : \mathbb{R}$ (functional determination)

$\beta_L : \mathbb{R}$ (cpd confidence)

- $(\mathcal{N}, \mathcal{V})$ is a set of variables

Definition (PDG)

$\mathcal{N}$: **Set** (node set)

$\mathcal{V} : \mathcal{N} \to \mathbf{Set}$ (node values)

$\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N} \times \mathit{Label}$ (edge set)

For $X \xrightarrow{L} Y \in \mathcal{E}$,

$\mathbf{p}_L : \mathcal{V}(X) \to \Delta\mathcal{V}(Y)$ (edge cpd)

$\alpha_L : \mathbb{R}$ (functional determination)

$\beta_L : \mathbb{R}$ (cpd confidence)

- $(\mathcal{N}, \mathcal{V})$ is a set of variables
- $(\mathcal{N}, \mathcal{E})$ is a multigraph

Definition (PDG)

$\mathcal{N}$: **Set** (node set)

$\mathcal{V} : \mathcal{N} \to \mathbf{Set}$ (node values)

$\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N} \times \mathit{Label}$ (edge set)

For $X \xrightarrow{L} Y \in \mathcal{E}$,

$\mathbf{p}_L : \mathcal{V}(X) \to \Delta\mathcal{V}(Y)$ (edge cpd)

$\alpha_L : \mathbb{R}$ (functional determination)

$\beta_L : \mathbb{R}$ (cpd confidence)

- $(\mathcal{N}, \mathcal{V})$ is a set of variables
- $(\mathcal{N}, \mathcal{E})$ is a multigraph
- $(\mathcal{N}, \mathcal{E}, \alpha)$, the qualitative data, forms a weighted multigraph.

Definition (PDG)

$\mathcal{N}$: **Set** (node set)

$\mathcal{V} : \mathcal{N} \to$ **Set** (node values)

$\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N} \times \mathit{Label}$ (edge set)

For $X \xrightarrow{L} Y \in \mathcal{E}$,

$\mathbf{p}_L : \mathcal{V}(X) \to \Delta\mathcal{V}(Y)$ (edge cpd)

$\alpha_L : \mathbb{R}$ (functional determination)

$\beta_L : \mathbb{R}$ (cpd confidence)

- $(\mathcal{N}, \mathcal{V})$ is a set of variables
- $(\mathcal{N}, \mathcal{E})$ is a multigraph
- $(\mathcal{N}, \mathcal{E}, \alpha)$, the qualitative data, forms a weighted multigraph.
- We call $(\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p})$ an *unweighted* PDG
  - and give it semantics as though $\alpha_L = \beta_L = 1$.

Definition (PDG)

$\mathcal{N}$: **Set** (node set)

$\mathcal{V} : \mathcal{N} \to$ **Set** (node values)

$\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N} \times \mathit{Label}$ (edge set)

For $X \xrightarrow{L} Y \in \mathcal{E}$,

$\mathbf{p}_{L} : \mathcal{V}(X) \to \Delta\mathcal{V}(Y)$ (edge cpd)

$\alpha_{L} : \mathbb{R}$ (functional determination)

$\beta_{L} : \mathbb{R}$ (cpd confidence)

- $(\mathcal{N}, \mathcal{V})$ is a set of variables
- $(\mathcal{N}, \mathcal{E})$ is a multigraph
- $(\mathcal{N}, \mathcal{E}, \alpha)$, the qualitative data, forms a weighted multigraph.
- We call $(\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p})$ an *unweighted* PDG
  - and give it semantics as though $\alpha_L = \beta_L = 1$.

Let **Mark** be the category of measurable spaces and Markov kernels.

Definition (PDG)

$\mathcal{N}$: **Set** (node set)

$\mathcal{V} : \mathcal{N} \to$ **Set** (node values)

$\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N} \times \mathit{Label}$ (edge set)

For $X \xrightarrow{L} Y \in \mathcal{E}$,

$\mathbf{p}_L : \mathcal{V}(X) \to \Delta\mathcal{V}(Y)$ (edge cpd)

$\alpha_L : \mathbb{R}$ (functional determination)

$\beta_L : \mathbb{R}$ (cpd confidence)

- $(\mathcal{N}, \mathcal{V})$ is a set of variables
- $(\mathcal{N}, \mathcal{E})$ is a multigraph
- $(\mathcal{N}, \mathcal{E}, \alpha)$, the qualitative data, forms a weighted multigraph.
- We call $(\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p})$ an *unweighted* PDG
  - and give it semantics as though $\alpha_L = \beta_L = 1$.

Let **Mark** be the category of measurable spaces and Markov kernels.

Equivalent Categorical Definition

An unweighted PDG is a functor $\langle \mathbf{p}, \mathcal{V} \rangle : \mathit{Paths}(\mathcal{N}, \mathcal{E}) \to \mathbf{Mark}$.
So a PDG is a *diagram* in **Mark**, in the usual mathematical sense.

What do you do with diagrams? Take limits / colimits.

$$\cdots \qquad X_1 \twoheadrightarrow X_2 \leftarrow X_3 \qquad \cdots$$

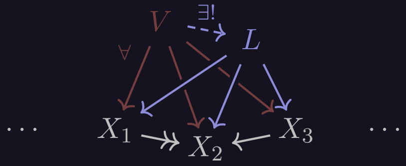
What do you do with diagrams? Take limits / colimits.

$$\begin{tikzcd}
 & & L \arrow[dl] \arrow[d] \arrow[dr] & & \\
\cdots & X_1 \arrow[r, two heads] & X_2 & X_3 \arrow[l] & \cdots
\end{tikzcd}$$

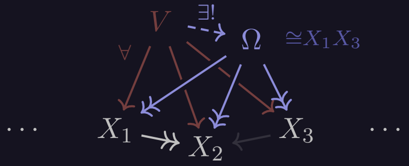
What do you do with diagrams? Take limits / colimits.

$$
\begin{tikzcd}
 & & V \arrow[dl, "\forall"'] \arrow[d] \arrow[dr] & L \arrow[dll] \arrow[dl] \arrow[d] & \\
\cdots & X_1 \arrow[r, two heads] & X_2 & X_3 \arrow[l] & \cdots
\end{tikzcd}
$$

What do you do with diagrams? Take limits / colimits.



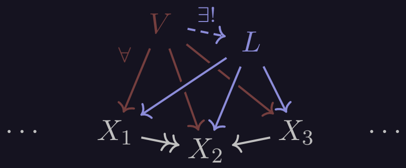
What do you do with diagrams? Take limits / colimits.



For the deterministic sub-PDG $m_{\det} \subseteq m$:

$$\lim m_{\det} = \left(\begin{array}{c}\text{natural}\\ \text{sample space}\end{array} \Omega, \quad \begin{array}{c}\text{random}\\ \text{variables}\end{array} \left\{ \tilde{X} : \Omega \to \mathcal{V}(X) \right\}_{X \in \mathcal{N}} \right)$$

What do you do with diagrams? Take limits / colimits.

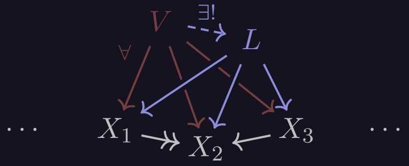


For the deterministic sub-PDG $m_{\text{det}} \subseteq m$:

$$\lim m_{\text{det}} = \left(\begin{array}{c}\text{natural}\\ \text{sample space}\end{array} \Omega, \quad \begin{array}{c}\text{random}\\ \text{variables}\end{array} \left\{ \tilde{X} : \Omega \to \mathcal{V}(X) \right\}_{X \in \mathcal{N}} \right)$$

In general: $$\lim m = \Big(\mathrm{Verts}(\underbrace{\mathbb{L}m}_{\substack{\text{Locally Consistent Polytope} \\ \text{(possible states of the Sum-Product algorithm)}}}), \quad \{ \text{variable marginals} \} \Big)$$

What do you do with diagrams? Take limits / colimits.



For the deterministic sub-PDG $\mathcal{m}_{\text{det}} \subseteq \mathcal{m}$:

$$\lim \mathcal{m}_{\text{det}} = \left(\begin{array}{c}\text{natural}\\ \text{sample space}\end{array} \ \Omega, \quad \begin{array}{c}\text{random}\\ \text{variables}\end{array} \left\{ \tilde{X} : \Omega \to \mathcal{V}(X) \right\}_{X \in \mathcal{N}} \right)$$

In general: $$\lim \mathcal{m} = \Big(\text{Verts}(\underbrace{\mathbb{L}\mathcal{m}}_{\substack{\text{Locally Consistent Polytope} \\ \text{(possible states of the Sum-Product algorithm)}}}), \quad \{ \text{variable marginals} \} \Big)$$

For a BN $\mathcal{B}$: $$\lim \mathcal{m}_{\mathcal{B}} = \left(\mathbb{1}, \ \left\{ \Pr_{\mathcal{B}}(X) \right\}_{X \in \mathcal{N}} \right)$$

Outline for Section 3

Semantics of PDGs

$\{\!\{m\}\!\} \subseteq \Delta\mathcal{V}(m)$

The set of joint distributions consistent with m;

Semantics of PDGs

$\{\!\{ m \}\!\} \subseteq \Delta \mathcal{V}(m)$

The set of joint distributions consistent with m;

$[\![m]\!]_\gamma : \Delta \mathcal{V}(m) \to \mathbb{R}$

A function (parameterized by $\gamma > 0$) that scores distributions by compatibility with m;

Semantics of PDGs

$\{\!\!\{ m \}\!\!\} \subseteq \Delta\mathcal{V}(m)$

The set of joint distributions consistent with m;

$[\![m]\!]_\gamma : \Delta\mathcal{V}(m) \to \mathbb{R}$

A function (parameterized by $\gamma > 0$) that scores distributions by compatibility with m;

$[\![m]\!]^*_\gamma \subseteq \Delta\mathcal{V}(m)$

The distribution(s) most compatible with m
(a singleton in many cases of interest);

Semantics of PDGs

$\{\!\{ m \}\!\} \subseteq \Delta\mathcal{V}(m)$

The set of joint distributions consistent with m;

$[\![m]\!]_\gamma : \Delta\mathcal{V}(m) \to \mathbb{R}$

A function (parameterized by $\gamma > 0$) that scores distributions by compatibility with m;

$[\![m]\!]^*_\gamma \subseteq \Delta\mathcal{V}(m)$

The distribution(s) most compatible with m
(a singleton in many cases of interest);

$\langle\!\langle m \rangle\!\rangle_\gamma : \mathbb{R}$

The best posssible compatibility of m with any distribution: the *inconsistency* of m

Semantics of PDGs

$\{\!\{m\}\!\} \subseteq \Delta\mathcal{V}(m)$

The set of joint distributions consistent with m;

$[\![m]\!]_\gamma : \Delta\mathcal{V}(m) \to \mathbb{R}$

A function (parameterized by $\gamma > 0$) that scores distributions by compatibility with m;

$[\![m]\!]^*_\gamma \subseteq \Delta\mathcal{V}(m)$

The distribution(s) most compatible with m
(a singleton in many cases of interest);

$\langle\!\langle m \rangle\!\rangle_\gamma : \mathbb{R}$

The best posssible compatibility of m with any distribution: the *inconsistency* of m

Semantics of PDGs

$\{\!\!\{ m \}\!\!\} \subseteq \Delta\mathcal{V}(m)$

The set of joint distributions consistent with m;

$[\![m]\!]_\gamma : \Delta\mathcal{V}(m) \to \mathbb{R}$

A function (parameterized by $\gamma > 0$) that scores distributions by compatibility with m;

$[\![m]\!]^*_\gamma \subseteq \Delta\mathcal{V}(m)$

The distribution(s) most compatible with m
(a singleton in many cases of interest);

$\langle\!\langle m \rangle\!\rangle_\gamma : \mathbb{R}$

The best posssible compatibility of m with any distribution: the *inconsistency* of m

Semantics of PDGs

$\{\!\{m\}\!\} \subseteq \Delta\mathcal{V}(m)$

The set of joint distributions consistent with m;

$[\![m]\!]_\gamma : \Delta\mathcal{V}(m) \to \mathbb{R}$

A function (parameterized by $\gamma > 0$) that scores distributions by compatibility with m;

$[\![m]\!]^*_\gamma \subseteq \Delta\mathcal{V}(m)$

The distribution(s) most compatible with m
(a singleton in many cases of interest);

$\langle\!\langle m \rangle\!\rangle_\gamma : \mathbb{R}$

The best posssible compatibility of m with any distribution: the *inconsistency* of m

THE SCORING FUNCTION

⟨ either finish overhaul or reinstate old slides ⟩

$$[\![\mathcal{M}]\!]_\gamma(\mu) := \mathit{Inc}_{\mathcal{M}}(\mu) + \gamma\ \mathit{IDef}_{\mathcal{M}}(\mu)$$

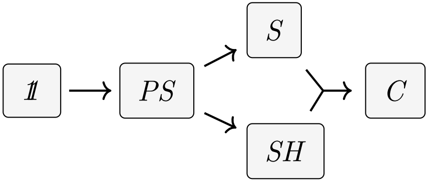
tradeoff parameter $\gamma \geq 0$

⟨ Emphasize entropy balance ⟩

Outline for Section 4

Capturing Bayesian Networks

For a BN $\mathcal{B}$ with N nodes and a vector $\beta \in \mathbb{R}^N$, let $\boldsymbol{m}_{\mathcal{B},\beta}$ be the PDG corresponding to $\mathcal{B}$, with $\alpha = \mathbf{1}$, and the given vector β of confidences.



Capturing Bayesian Networks

For a BN $\mathcal{B}$ with N nodes and a vector $\beta \in \mathbb{R}^N$, let $\boldsymbol{m}_{\mathcal{B},\beta}$ be the PDG corresponding to $\mathcal{B}$, with $\alpha = \mathbf{1}$, and the given vector β of confidences.

Theorem (*BNs are PDGs*)

If $\mathcal{B}$ is a BN and $\Pr_{\mathcal{B}}$ is the distribution it specifies, then for all $\gamma > 0$ and all vectors β,

$$[\![\boldsymbol{m}_{\mathcal{B},\beta}]\!]^*_\gamma = \{\Pr_{\mathcal{B}}\}, \quad \textit{and thus} \quad [\![\boldsymbol{m}_{\mathcal{B},\beta}]\!]^* = \Pr_{\mathcal{B}}\,.$$

Capturing Bayesian Networks

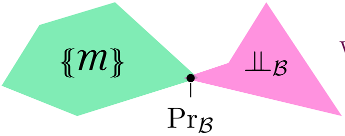
For a BN $\mathcal{B}$ with N nodes and a vector $\beta \in \mathbb{R}^N$, let $\boldsymbol{m}_{\mathcal{B},\beta}$ be the PDG corresponding to $\mathcal{B}$, with $\alpha = \mathbf{1}$, and the given vector β of confidences.

Theorem (*BNs are PDGs*)

If $\mathcal{B}$ is a BN and $\Pr_{\mathcal{B}}$ is the distribution it specifies, then for all $\gamma > 0$ and all vectors β,

$$[\![\boldsymbol{m}_{\mathcal{B},\beta}]\!]^*_\gamma = \{\Pr\nolimits_{\mathcal{B}}\}, \quad \textit{and thus} \quad [\![\boldsymbol{m}_{\mathcal{B},\beta}]\!]^* = \Pr\nolimits_{\mathcal{B}}\,.$$

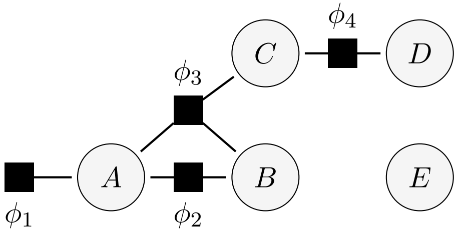
space of distributions consistent with $\boldsymbol{m}_{\mathcal{B}}$ (which minimize *Inc*)



space of distributions with independencies of $\mathcal{B}$ (which can be shown to minimize *IDef*)

$\langle$ maximum entropy result for BNs $\rangle$

Factor Graphs



You need to add some intuition here.

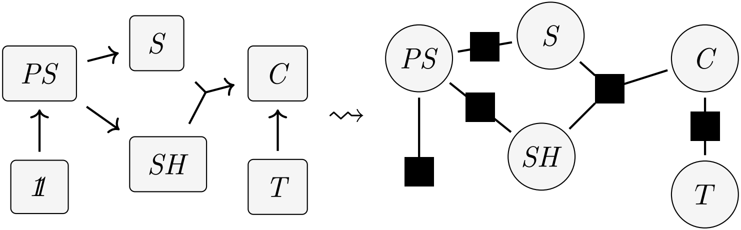
Definition

A *factor graph* Φ is a set of random variables $\mathcal{X} = \{X_i\}$ and *factors* $\{\phi_J : \mathcal{V}(X_J) \to \mathbb{R}_{\geq 0}\}_{J \in \mathcal{J}}$, where $X_J \subseteq \mathcal{X}$; define

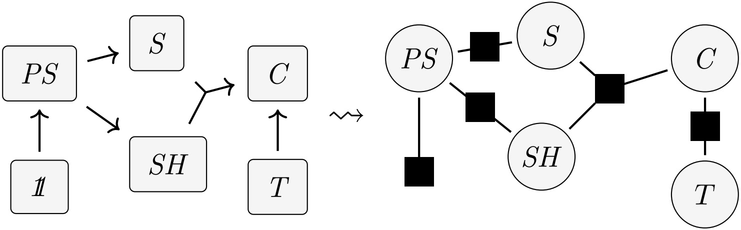
$$\Pr_\Phi(\vec{x}) = \frac{1}{Z_\Phi} \prod_{J \in \mathcal{J}} \phi_J(\vec{x}_J),$$

where Z_Φ is the normalization constant.

PDGs as Factor Graphs

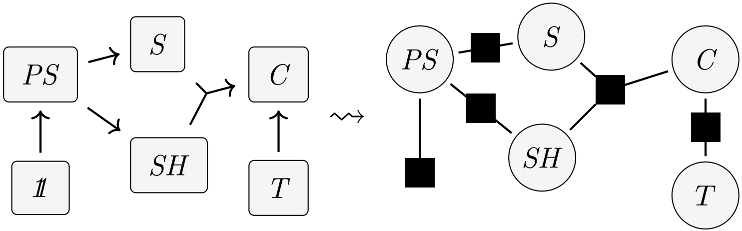


PDGs as Factor Graphs



The cpds of a PDG are essentially factors. Are the semantics different?

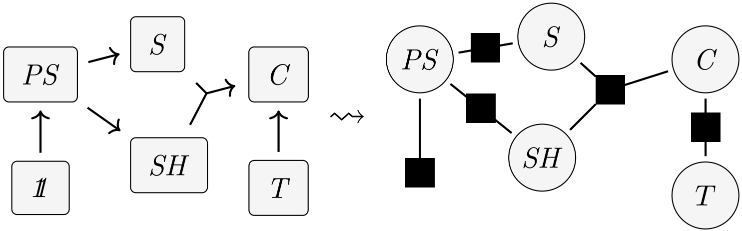
PDGs as Factor Graphs



The cpds of a PDG are essentially factors. Are the semantics different?

Not for $\gamma = 1$.

PDGs as Factor Graphs



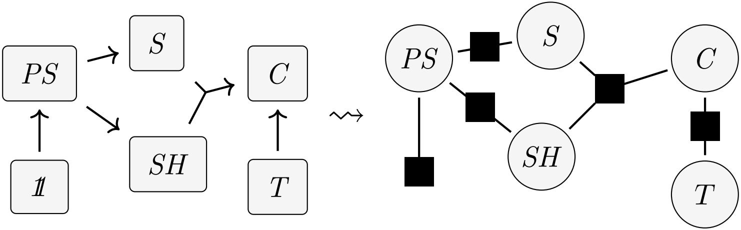
The cpds of a PDG are essentially factors. Are the semantics different?

Not for $\gamma = 1$.

Theorem

$[\![\mathcal{M}]\!]_1^* = \Pr_{\Phi_{\mathcal{M}}}$ *for all unweighted PDGs* $\mathcal{M}$.

PDGs as Factor Graphs



The cpds of a PDG are essentially factors. Are the semantics different?

Not for $\gamma = 1$.

Theorem

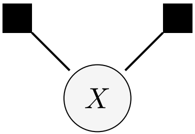
$[\![\mathcal{M}]\!]_1^* = \Pr_{\Phi_{\mathcal{M}}}$ *for all unweighted PDGs* $\mathcal{M}$.

Theorem

For all unweighted PDGs $\mathcal{M}$ *and non-negative vectors* $\mathbf{v}$ *over the edges of* $\mathcal{M}$, *and all* $\gamma > 0$, *we have that* $[\![(\mathcal{M}, \mathbf{v}, \gamma\mathbf{v})]\!]_\gamma = \gamma\, \mathit{GFE}_{(\Phi_{\mathcal{M}}, \mathbf{v})}$; *consequently,* $[\![(\mathcal{M}, \mathbf{v}, \gamma\mathbf{v})]\!]_\gamma^* = \{\Pr_{(\Phi_{\mathcal{M}}, \mathbf{v})}\}$.

An Important Difference between PDGs and Factor Graphs

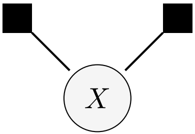
$$m := \begin{tikzcd} \mathbb{1} \arrow[d, bend right, "q"'] \arrow[d, bend left, "p"] \\ X \end{tikzcd}$$



$$=: \Phi$$

An Important Difference between PDGs and Factor Graphs

$$\boldsymbol{m} := \begin{tikzcd} \boxed{\mathbb{1}} \arrow[d, bend right, "q"'] \arrow[d, bend left, "p"] \\ \boxed{X} \end{tikzcd}$$

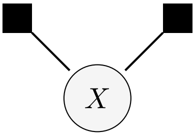


$=: \Phi$

- If $p = q$, then $[\![\boldsymbol{m}]\!]^* = p = q \ldots$

An Important Difference between PDGs and Factor Graphs

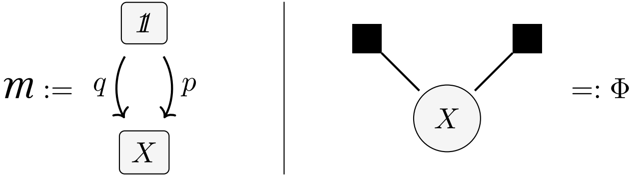
$$\boldsymbol{m} := \begin{tikzcd} \boxed{\mathbb{1}} \arrow[d, bend right, "q"'] \arrow[d, bend left, "p"] \\ \boxed{X} \end{tikzcd}$$



$=: \Phi$

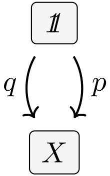
- If $p = q$, then $[\![\boldsymbol{m}]\!]^* = p = q \ldots$
- $\ldots$ but $\Pr_\Phi \propto p^2$

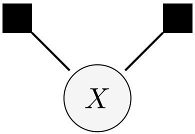
An Important Difference between PDGs and Factor Graphs



- If $p = q$, then $[\![m]\!]^* = p = q \ldots$
- $\ldots$ but $\Pr_\Phi \propto p^2$
- Individual factors have *no probabilistic meaning,*

An Important Difference between PDGs and Factor Graphs

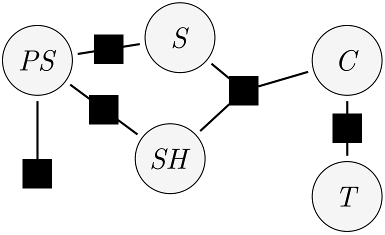
$\mathcal{m} :=$ 



$=: \Phi$

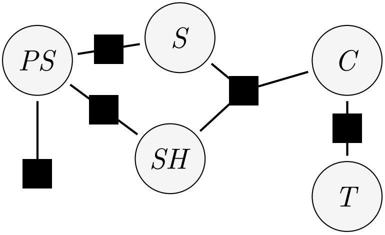
- If $p = q$, then $[\![\mathcal{m}]\!]^* = p = q \ldots$
- $\ldots$ but $\Pr_\Phi \propto p^2$
- Individual factors have *no probabilistic meaning*,
- a factor graph can fail to normalize, in which case it has no global semantics either.

Factor Graphs as PDGs

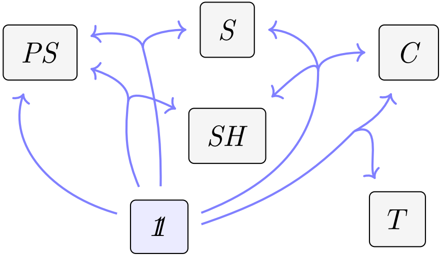


$\rightsquigarrow$

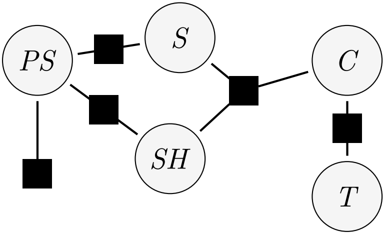
Factor Graphs as PDGs



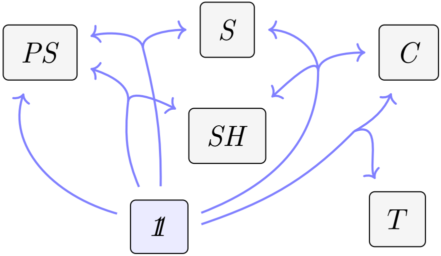
$\rightsquigarrow$



Factor Graphs as PDGs



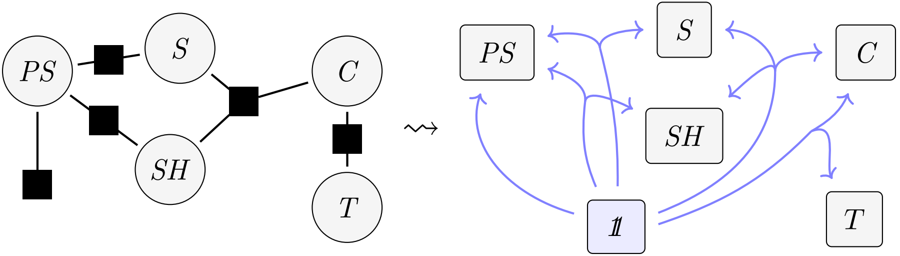
$\rightsquigarrow$



Theorem

$\Pr_\Phi = [\![\mathcal{M}_\Phi]\!]^*_1$ *for all factor graphs* Φ.

Factor Graphs as PDGs



Theorem

$\Pr_\Phi = [\![\mathcal{n}_\Phi]\!]_1^*$ *for all factor graphs* Φ.

Theorem

For all weighted factor graphs $\Psi = (\Phi, \theta)$ *and all* $\gamma > 0$, *we have that* $\mathit{GFE}_\Psi = {}^1\!/_\gamma [\![\mathcal{m}_{\Psi,\gamma}]\!]_\gamma + C$ *for some constant* C, *so* $\Pr_\Psi$ *is the unique element of* $[\![\mathcal{m}_{\Psi,\gamma}]\!]_\gamma^*$.

⟨ `Add theorem:` $\log Z_\Phi = \langle\!\langle \mathcal{n}_\Phi \rangle\!\rangle$ ⟩

I suspect that you'll lose your audience with this

Letting $x^{\mathbf{w}}$ and $y^{\mathbf{w}}$ denote the values of X and Y, respectively, in $\mathbf{w} \in \mathcal{V}(\mathcal{M})$, we have

$$[\![\mathcal{M}]\!](\mu) = \mathop{\mathbb{E}}_{\mathbf{w}\sim\mu} \Bigg\{ \sum_{X \xrightarrow{L} Y} \Bigg[\overbrace{\beta_L \log \frac{1}{\mathbf{p}_L(y^{\mathbf{w}} | x^{\mathbf{w}})}}^{\text{log likelihood / cross entropy}} + \underbrace{(\alpha_L \gamma - \beta_L) \log \frac{1}{\mu(y^{\mathbf{w}} | x^{\mathbf{w}})}}_{\text{local regularization } (\beta_L > \alpha_L \gamma)} \Bigg] - \underbrace{\gamma \log \frac{1}{\mu(\mathbf{w})}}_{\text{global regularization}} \Bigg\}.$$

Outline for Section 5

Inference via Inconsistency Reduction

Conditioning as inconsistency resolution.
To condition on $Y = y$, in $\mathcal{M}$, simply add the edge $\mathbb{1} \xrightarrow{\delta_y} Y$ to get $\mathcal{M}_{Y=y}$.
Then $[\![\mathcal{M}_{Y=y}]\!]^* = [\![\mathcal{M}]\!]^* \mid (Y = y)$.

Add some words for what this is saying (e.g., remind the reader of what [[\M]]* is).Ar

Conditioning as inconsistency resolution.
To condition on $Y = y$, in $\mathcal{M}$, simply add the edge $\mathbb{1} \xrightarrow{\delta_y} Y$ to get $\mathcal{M}_{Y=y}$.
Then $[\![\mathcal{M}_{Y=y}]\!]^* = [\![\mathcal{M}]\!]^* \mid (Y = y)$.

Querying $\Pr(Y \mid X)$ **in a PDG** $\mathcal{M}$**.**

- We can add $X \xrightarrow{p} Y$ to $\mathcal{M}$ with a cpt p, to get $\mathcal{M}^{+p}$.

Text

⟨ Convexity Result ⟩

⟨ Hardness Result ⟩

Inference via Inconsistency Reduction

Conditioning as inconsistency resolution.
To condition on $Y = y$, in $\mathcal{M}$, simply add the edge $\mathbb{1} \xrightarrow{\delta_y} Y$ to get $\mathcal{M}_{Y=y}$.
Then $[\![\mathcal{M}_{Y=y}]\!]^* = [\![\mathcal{M}]\!]^* \mid (Y = y)$.

Querying $\Pr(Y \mid X)$ **in a PDG** $\mathcal{M}$**.**

- We can add $X \xrightarrow{p} Y$ to $\mathcal{M}$ with a cpt p, to get $\mathcal{M}^{+p}$.
- The choice of cpd p that minimizes the inconsistency of $\mathcal{M}^{+p}$ (which is strongly convex and smooth in p) is $[\![\mathcal{M}]\!]^*(Y \mid X)$,

⟨ Convexity Result ⟩
⟨ Hardness Result ⟩

Inference via Inconsistency Reduction

Conditioning as inconsistency resolution.
To condition on $Y = y$, in $\mathcal{m}$, simply add the edge $\mathbb{1} \xrightarrow{\delta_y} Y$ to get $\mathcal{m}_{Y=y}$.
Then $[\![\mathcal{m}_{Y=y}]\!]^* = [\![\mathcal{m}]\!]^* \mid (Y = y)$.

Querying $\Pr(Y \mid X)$ **in a PDG** $\mathcal{m}$**.**

- We can add $X \xrightarrow{p} Y$ to $\mathcal{m}$ with a cpt p, to get $\mathcal{m}^{+p}$.
- The choice of cpd p that minimizes the inconsistency of $\mathcal{m}^{+p}$ (which is strongly convex and smooth in p) is $[\![\mathcal{m}]\!]^*(Y \mid X)$,
- so oracle access to inconsistency yields fast inference by gradient descent.

Make this a sub-bullet

⟨ Convexity Result ⟩

⟨ Hardness Result ⟩

Outline for Section 6

Inconsistency: The Universal Loss

- Fruitful to cast AI as optimization. But what to optimize?

Inconsistency: The Universal Loss

- Fruitful to cast AI as optimization. But what to optimize?
  - `Cross Entropy`, `Square Loss`, `Accuracy`, ...

Inconsistency: The Universal Loss

- Fruitful to cast AI as optimization. But what to optimize?
  - `Cross Entropy`, `Square Loss`, `Accuracy`, ...
  - choice is made by instinct, tradition, or pragmatics, which makes results difficult to motivate, and vulnerable to overfitting.

Inconsistency: The Universal Loss

- Fruitful to cast AI as optimization. But what to optimize?
  - `Cross Entropy`, `Square Loss`, `Accuracy`, ...
  - choice is made by instinct, tradition, or pragmatics, which makes results difficult to motivate, and vulnerable to overfitting.
- Choice of *model* admits more principled discussion.

Inconsistency: The Universal Loss

- Fruitful to cast AI as optimization. But what to optimize?
  - `Cross Entropy`, `Square Loss`, `Accuracy`, ...
  - choice is made by instinct, tradition, or pragmatics, which makes results difficult to motivate, and vulnerable to overfitting.
- Choice of *model* admits more principled discussion.
  - and a possibly-inconsistent model comes with a natural objective.

This needs more intuition. What's the model doing? Why might it be inconsistent?

Inconsistency: The Universal Loss

- Fruitful to cast AI as optimization. But what to optimize?
  - `Cross Entropy`, `Square Loss`, `Accuracy`, ...
  - choice is made by instinct, tradition, or pragmatics, which makes results difficult to motivate, and vulnerable to overfitting.
- Choice of *model* admits more principled discussion.
  - and a possibly-inconsistent model comes with a natural objective.

Inconsistency: The Universal Loss

- Fruitful to cast AI as optimization. But what to optimize?
  - `Cross Entropy`, `Square Loss`, `Accuracy`, ...
  - choice is made by instinct, tradition, or pragmatics, which makes results difficult to motivate, and vulnerable to overfitting.
- Choice of *model* admits more principled discussion.
  - and a possibly-inconsistent model comes with a natural objective.

Surprising Result

Most standard objectives arise naturally as the inconsistency of the obvious PDG describing the situation.

obvious -> natural

Inconsistency: The Universal Loss

- Fruitful to cast AI as optimization. But what to optimize?
  - `Cross Entropy`, `Square Loss`, `Accuracy`, ...
  - choice is made by instinct, tradition, or pragmatics, which makes results difficult to motivate, and vulnerable to overfitting.
- Choice of *model* admits more principled discussion.
  - and a possibly-inconsistent model comes with a natural objective.

Surprising Result

Most standard objectives arise naturally as the inconsistency of the obvious PDG describing the situation.

Pedagogical Bonus

An intuitive visual language for reasoning about ineuqalities

This seems misplaced. “Inequalities” comes out of the blue.

Surprise as Inconsistency

Proposition

Consider a distribution over X with mass function $p(X)$. The surprise (or information content) $\mathrm{I}_p(x) := -\log p(X{=}x)$ at seeing a sample x is the inconsistency of the pdg containing p and the event $X = x$, i.e.,

$$\mathrm{I}_p(x) = \log \frac{1}{p(X{=}x)} = \left\langle\!\!\left\langle \xrightarrow{\;p\;} \boxed{X} \overset{X=x}{\twoheadleftarrow\!\!-\!\!-} \right\rangle\!\!\right\rangle.$$

Where did the << >> notation come from?

Surprise as Inconsistency

Proposition

Consider a distribution over X with mass function $p(X)$. The surprise (or information content) $\mathrm{I}_p(x) := -\log p(X{=}x)$ at seeing a sample x is the inconsistency of the pdg containing p and the event $X = x$, i.e.,

$$\mathrm{I}_p(x) = \log \frac{1}{p(X{=}x)} = \left\langle\!\!\left\langle \xrightarrow{p} \boxed{X} \twoheadleftarrow^{X=x} \right\rangle\!\!\right\rangle.$$

- PDG semantics just so happen to give the standard meaure of compatibility between a sample and distribution.

Surprise as Inconsistency

Proposition

Consider a distribution over X with mass function $p(X)$. The surprise (or information content) $\mathrm{I}_p(x) := -\log p(X{=}x)$ at seeing a sample x is the inconsistency of the pdg containing p and the event $X = x$, i.e.,

$$\mathrm{I}_p(x) = \log \frac{1}{p(X{=}x)} = \left\langle\!\!\left\langle \xrightarrow{p} \boxed{X} \twoheadleftarrow\!\!\xleftarrow{X=x} \right\rangle\!\!\right\rangle.$$

- PDG semantics just so happen to give the standard meaure of compatibility between a sample and distribution.
- Known as "surprise", a particular kind of internal conflict.

Variations: Surprise as Inconsistency

Proposition (marginal information as inconsistency)

If $p(X, Z)$ is a joint distribution, the (marginal) information of the (partial) observation $X = x$ is given by

$$\mathrm{I}_p(x) = \log \frac{1}{p(x)} = \left\langle\!\!\left\langle\; Z \overset{p}{\longleftarrow\!\!\longrightarrow} X \overset{x}{\twoheadleftarrow} \;\right\rangle\!\!\right\rangle.$$

While you may be used to all these funny symbols, the listener won't be. You have to be careful not to go overboard here.

Variations: Surprise as Inconsistency

Proposition (marginal information as inconsistency)

If $p(X, Z)$ is a joint distribution, the (marginal) information of the (partial) observation $X = x$ is given by

$$\mathrm{I}_p(x) = \log \frac{1}{p(x)} = \left\langle\!\!\left\langle\; \boxed{Z} \overset{p}{\swarrow\!\searrow} \boxed{X} \overset{x}{\twoheadleftarrow} \;\right\rangle\!\!\right\rangle.$$

Proposition (cross entropy as inconsistency)

Given a dataset $\underline{\mathbf{x}}$, the cross entropy $\mathrm{CE}(p, \underline{\mathbf{x}}) := -\frac{1}{|\underline{\mathbf{x}}|} \sum_{x \in \underline{\mathbf{x}}} \log p(x)$ is the inconsistency of the PDG containing p and the data distribution $\Pr_{\underline{\mathbf{x}}}$, plus the entropy of the data distribution (constant in p). That is,

I still object to the !, and I assure you the listener will have no clue of what it means.

$$\mathrm{CE}(p; \underline{\mathbf{x}}) = \left\langle\!\!\left\langle\; \boxed{Z} \overset{p}{\swarrow\!\searrow} \boxed{X} \xleftarrow{\Pr_{\underline{\mathbf{x}}}!} \;\right\rangle\!\!\right\rangle + \mathrm{H}(\Pr_{\underline{\mathbf{x}}}).$$

You're now officiallly overboard.

Proposition (Accuracy as Inconsistency)

Consider a predictor $h : X \to Y$ for true labels $f : X \to Y$, and a distribution $D(X)$. The inconsistency of believing all three is

What is accuracy?

$$\left\langle\!\!\left\langle \begin{tikzcd} {} \arrow[r, "D^{(\beta)}"] & \boxed{X} \arrow[r, bend left, "h"] \arrow[r, bend right, "f"'] & \boxed{Y} \end{tikzcd} \right\rangle\!\!\right\rangle = -\beta \log\Big(\mathrm{accuracy}_{f,D}(h)\Big) = \beta\ \mathrm{I}_D[f = h].$$

Proposition (Mean Square Error as Inconsistency)

$$\left\langle\!\!\left\langle \begin{tikzcd} {} \arrow[r, "D!"] & X \arrow[r, bend left, "\mathcal{N}(f(x),1)"] \arrow[r, bend right, "\mathcal{N}(g(x),1)"'] & Y \end{tikzcd} \right\rangle\!\!\right\rangle = \left\langle\!\!\left\langle \begin{tikzcd} & & \mu_f \arrow[dr, "\mathcal{N}_1"] & \\ {} \arrow[r, "D!"] & X \arrow[ur, two heads, "f"] \arrow[dr, two heads, "h"'] & & Y \\ & & \mu_h \arrow[ur, "\mathcal{N}_1"'] & \end{tikzcd} \right\rangle\!\!\right\rangle$$

$$= \mathbb{E}_D \Big(f(X) - h(X) \Big)^2 =: \mathrm{MSE}(f, h)$$

where $\mathcal{N}_1 = \mathcal{N}(-, 1)$ is the normal distribution with unit variance, and mean equal to its argument.

Proposition (Regularizers as priors)

Suppose you believe $Y \sim f_\theta(Y)$,

That is,

$$\left\langle\!\!\left\langle \quad \boxed{\Theta} \xrightarrow{f} \boxed{Y} \quad \right\rangle\!\!\right\rangle$$

What does this have to do with belief?
I'm lost.

Proposition (Regularizers as priors)

Suppose you believe $Y \sim f_\theta(Y)$, have a prior $p(\theta)$,

That is,

$$\left\langle\!\!\left\langle \xrightarrow{p^{(\beta)}} \boxed{\Theta} \xrightarrow{f} \boxed{Y} \right\rangle\!\!\right\rangle$$

Proposition (Regularizers as priors)

Suppose you believe $Y \sim f_\theta(Y)$, have a prior $p(\theta)$, and have an empirical distribution $D(Y)$ which you trust.

That is,

$$\left\langle\!\!\left\langle \xrightarrow{p^{(\beta)}} \boxed{\Theta} \xrightarrow{f} \boxed{Y} \xleftarrow{D!} \right\rangle\!\!\right\rangle$$

Proposition (Regularizers as priors)

Suppose you believe $Y \sim f_\theta(Y)$, have a prior $p(\theta)$, and have an empirical distribution $D(Y)$ which you trust. Then the inconsistency of also believing $\Theta = \theta_0$ is

That is,

$$\left\langle\!\!\left\langle \begin{array}{c} \xrightarrow{p^{(\beta)}} \\ \xrightarrow[\theta_0]{} \end{array} \boxed{\Theta} \xrightarrow{f} \boxed{Y} \xleftarrow{D!} \right\rangle\!\!\right\rangle =$$

Proposition (Regularizers as priors)

Suppose you believe $Y \sim f_\theta(Y)$, have a prior $p(\theta)$, and have an empirical distribution $D(Y)$ which you trust.Then the inconsistency of also believing $\Theta = \theta_0$ is the regularized-cross entropy loss, and controlled by the strength β_p of the prior. That is,

$$\left\langle\!\!\left\langle \begin{array}{c} \xrightarrow{p^{(\beta)}} \\ \xrightarrow[\theta_0]{} \end{array} \!\!\boxed{\Theta} \xrightarrow{f} \boxed{Y} \xleftarrow{D!} \right\rangle\!\!\right\rangle = \mathop{\mathbb{E}}_{y\sim D}\left[\log\frac{1}{f(y \mid \theta_0)}\right] + \beta \log\frac{1}{p(\theta_0)} - \mathrm{H}(D)$$

Proposition (Regularizers as priors)

Suppose you believe $Y \sim f_\theta(Y)$, have a prior $p(\theta)$, and have an empirical distribution $D(Y)$ which you trust.Then the inconsistency of also believing $\Theta = \theta_0$ isthe regularized-cross entropy loss, and controlled by the strength β_p of the prior. That is,

$$\left\langle\!\!\left\langle \begin{array}{c} \xrightarrow{p^{(\beta)}} \\ \xrightarrow{\theta_0} \end{array} \boxed{\Theta} \xrightarrow{f} \boxed{Y} \xleftarrow{D!} \right\rangle\!\!\right\rangle = \mathop{\mathbb{E}}_{y\sim D}\left[\log \frac{1}{f(y \mid \theta_0)}\right] + \beta \log \frac{1}{p(\theta_0)} - \mathrm{H}(D)$$

Using a (discretized) unit gaussian as a prior, $p(\theta) = \frac{1}{k}\exp(-\frac{1}{2}\theta^2)$ for a normaization constant k, the RHS becomes

$$\underbrace{\mathop{\mathbb{E}}_{D}\left[\log \frac{1}{f(Y \mid \theta_0)}\right]}_{\substack{\text{Cross entropy loss of } f_\theta \text{ w.r.t. } D \\ \text{(data-fit cost of } \theta_0\text{)}}} \quad + \quad \underbrace{\frac{\beta}{2}\theta_0^2}_{\substack{\ell_2 \text{ regularizer} \\ \text{(complexity cost of } \theta_0\text{)}}} \quad \underbrace{+\beta \log k - \mathrm{H}(D)}_{\text{constant in } f \text{ and } \theta_0}.$$

Variational Auto-Encoders, Take 1

- Structure consists of two networks:

Variational Auto-Encoders, Take 1

- Structure consists of two networks:

  $e(Z \mid X)$: encodes X's in a (small) latent space Z;

Encodes in what way? What's a latent space. I'm lost.

Variational Auto-Encoders, Take 1

- Structure consists of two networks:

  $e(Z \mid X)$: encodes X's in a (small) latent space Z;

  $d(X \mid Z)$: generate samples of X from Z.

Variational Auto-Encoders, Take 1

- Structure consists of two networks:

  $e(Z \mid X)$: encodes X's in a (small) latent space Z;
  $d(X \mid Z)$: generate samples of X from Z.

- Objective:

Variational Auto-Encoders, Take 1

- Structure consists of two networks:

  $e(Z \mid X)$: encodes X's in a (small) latent space Z;
  $d(X \mid Z)$: generate samples of X from Z.

- Objective:
  - Want to minimize "reconstruction error" for each x

$$\mathrm{Rec}(x) = - \mathop{\mathbb{E}}_{z \sim e|x} \log d(x \mid z)$$

Variational Auto-Encoders, Take 1

- Structure consists of two networks:
  $e(Z \mid X)$: encodes X's in a (small) latent space Z;
  $d(X \mid Z)$: generate samples of X from Z.
- Objective:
  - Want to minimize "reconstruction error" for each x
    $$\mathrm{Rec}(x) = - \mathop{\mathbb{E}}_{z \sim e|x} \log d(x \mid z)$$
  - you also have a prior $p(Z)$, which you want to match.

Variational Auto-Encoders, Take 1

- Structure consists of two networks:
  $e(Z \mid X)$: encodes X's in a (small) latent space Z;
  $d(X \mid Z)$: generate samples of X from Z.
- Objective:
  - Want to minimize "reconstruction error" for each x
    $$\mathrm{Rec}(x) = - \mathop{\mathbb{E}}_{z \sim e|x} \log d(x \mid z)$$
  - you also have a prior $p(Z)$, which you want to match.
  - Together, maximize $\mathrm{ELBO}_{p,e,d}(x) :=$
    $$-\boldsymbol{D}(e(Z|x) \,\|\, p(Z)) - \mathrm{Rec}(x) = \mathop{\mathbb{E}}_{z \sim e|x}\left[\log \frac{p(z)d(x \mid z)}{e(z \mid x)}\right]$$

What does this expression represent? Why should you want to maximize it? I continue to be lost.

Variational Auto-Encoders, Take 1

- Structure consists of two networks:

  $e(Z \mid X)$: encodes X's in a (small) latent space Z;
  $d(X \mid Z)$: generate samples of X from Z.

- Objective:
  - Want to minimize "reconstruction error" for each x

    $$\mathrm{Rec}(x) = - \mathop{\mathbb{E}}_{z \sim e|x} \log d(x \mid z)$$

  - you also have a prior $p(Z)$, which you want to match.
  - Together, maximize $\mathrm{ELBO}_{p,e,d}(x) :=$

$$-\boldsymbol{D}(e(Z|x) \parallel p(Z)) - \mathrm{Rec}(x) = \mathop{\mathbb{E}}_{z \sim e|x} \left[\log \frac{p(z) d(x \mid z)}{e(z \mid x)} \right] \leq \log \Pr_{pd}(x)$$

Variational Auto-Encoders, Take 1

- Structure consists of two networks:

  $e(Z \mid X)$: encodes X's in a (small) latent space Z;
  $d(X \mid Z)$: generate samples of X from Z.

- Objective:
  - Want to minimize "reconstruction error" for each x

    $$\mathrm{Rec}(x) = - \mathop{\mathbb{E}}_{z \sim e|x} \log d(x \mid z)$$

  - you also have a prior $p(Z)$, which you want to match.
  - Together, maximize $\mathrm{ELBO}_{p,e,d}(x) :=$

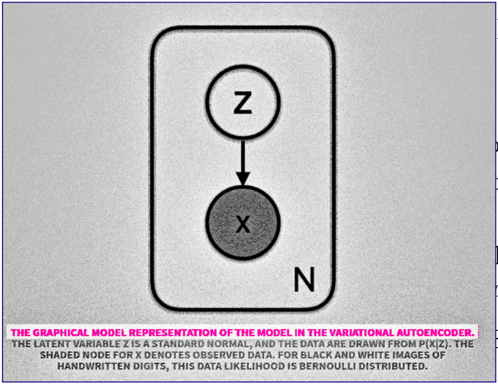
$$-\boldsymbol{D}(e(Z|x) \parallel p(Z)) - \mathrm{Rec}(x) = \mathop{\mathbb{E}}_{z \sim e|x}\left[\log \frac{p(z)d(x \mid z)}{e(z \mid x)}\right] \leq \log \Pr_{pd}(x)$$

Note: not a graphical model (desipte shoehorning attempts)

I have no idea why I should think it's a graphical model or what this note is referring to.

Variational Auto-Encoders, Take 1

- Structure consists of two networks:



l) latent space Z;

from Z.

on error" for each x

og $d(x \mid z)$

h you want to match.

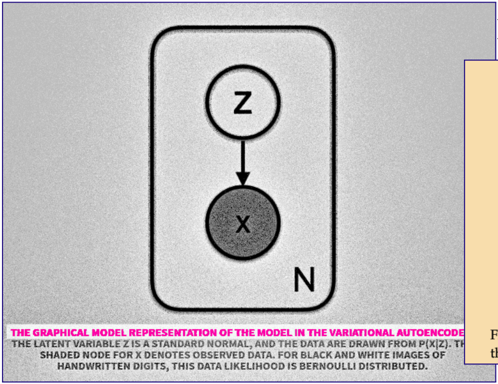
$x) :=$

$$\mathbb{E}_{e|x}\left[\log \frac{p(z)d(x \mid z)}{e(z \mid x)}\right] \leq \log \Pr_{pd}(x)$$

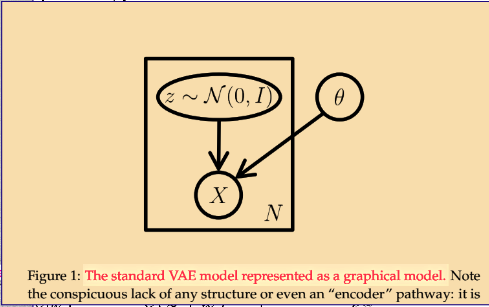
Note: not a graphical model (desipte shoehorning attempts)

Variational Auto-Encoders, Take 1

- Structure consists of two networks:



l) latent space Z;

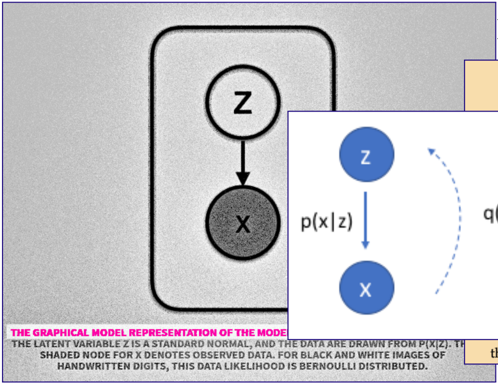


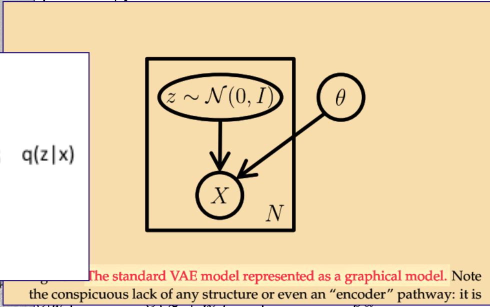
Note: not a graphical model (desipte shoehorning attempts)

Variational Auto-Encoders, Take 1

- Structure consists of two networks:

l) latent space Z;

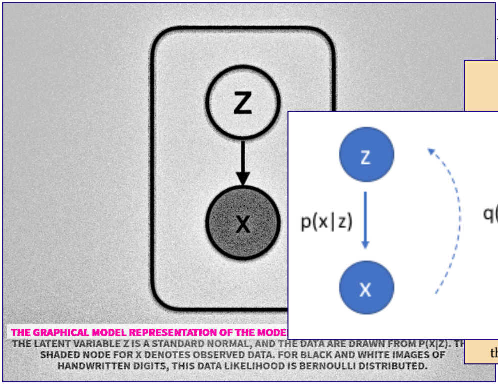




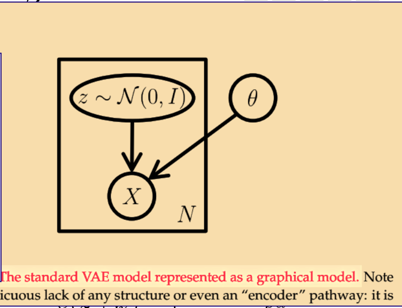
Note: not a graphical model (desipte shoehorning attempts)

Variational Auto-Encoders, Take 1

- Structure consists of two networks:



l) latent space Z;

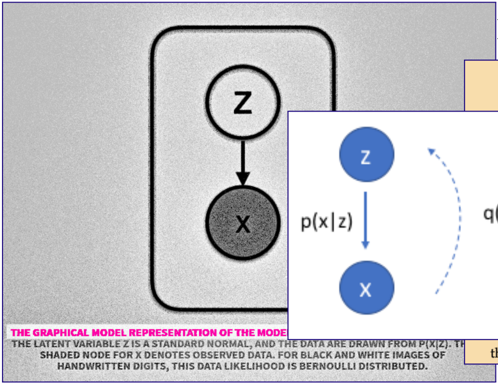


Note: not a graphical model (desipte shoehorning attempts)

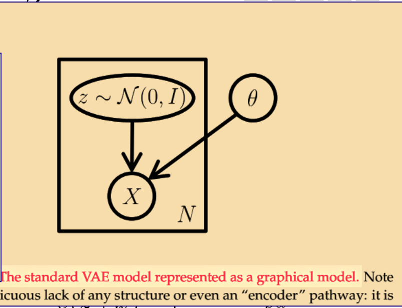
- $e(Z \mid X)$ has same target as $p(Z)$, so BN doesn't work

Variational Auto-Encoders, Take 1

- Structure consists of two networks:



l) latent space Z;



Note: not a graphical model (desipte shoehorning attempts)

- $e(Z \mid X)$ has same target as $p(Z)$, so BN doesn't work
- The heart of the VaE is not its structure, but its objective.

Variational Auto-Encoders, Take 2

- Structure:

$$\left\langle\!\!\left\langle \quad \boxed{Z} \quad \boxed{X} \quad \right\rangle\!\!\right\rangle = \mathrm{ELBO}_{p,e,d}(x)$$

Variational Auto-Encoders, Take 2

- Structure:

  $e(Z \mid X)$: encodes X in a latent space Z;

$$\left\langle\!\!\left\langle\ \begin{tikzcd} Z & X \arrow[l, bend left, "e"] \end{tikzcd}\ \right\rangle\!\!\right\rangle = \mathrm{ELBO}_{p,e,d}(x)$$

Variational Auto-Encoders, Take 2

- Structure:

  $e(Z \mid X)$: encodes X in a latent space Z;
  $d(X \mid Z)$: generate samples of X from Z.

$$\left\langle\!\!\left\langle \begin{tikzcd} Z \arrow[r, bend left, "d"] & X \arrow[l, bend left, "e"] \end{tikzcd} \right\rangle\!\!\right\rangle = \mathrm{ELBO}_{p,e,d}(x)$$

Variational Auto-Encoders, Take 2

- Structure:

  $e(Z \mid X)$: encodes X in a latent space Z;
  $d(X \mid Z)$: generate samples of X from Z.
  $p(Z)$: a prior over from Z.

$$\left\langle\!\!\left\langle \begin{tikzcd} {} \arrow[r, "p"] & Z \arrow[r, bend left, "d"] & X \arrow[l, bend left, "e"] \end{tikzcd} \right\rangle\!\!\right\rangle = \mathrm{ELBO}_{p,e,d}(x)$$

Variational Auto-Encoders, Take 2

- Structure:
  $e(Z \mid X)$: encodes X in a latent space Z;
  $d(X \mid Z)$: generate samples of X from Z.
  $p(Z)$: a prior over from Z.
- want to do a gradient step for a specific x.

I'm totally lost.

$$\left\langle\!\!\left\langle \begin{tikzcd} {} \arrow[r, "p"] & \boxed{Z} \arrow[r, bend left, "d"] & \boxed{X} \arrow[l, bend left, "e!"] & {} \arrow[l, twoheadrightarrow, "x"'] \end{tikzcd} \right\rangle\!\!\right\rangle = \mathrm{ELBO}_{p,e,d}(x)$$

Variational Auto-Encoders, Take 2

- Structure:
  $e(Z \mid X)$: encodes X in a latent space Z;
  $d(X \mid Z)$: generate samples of X from Z.
  $p(Z)$: a prior over from Z.
- want to do a gradient step for a specific x.

Objective function is free:

$$\left\langle\!\!\left\langle \begin{tikzcd} {} \arrow[r, "p"] & \boxed{Z} \arrow[r, bend left, "d"] & \boxed{X} \arrow[l, bend left, "e!"] & {} \arrow[l, two heads, "x"'] \end{tikzcd} \right\rangle\!\!\right\rangle = \mathrm{ELBO}_{p,e,d}(x)$$

A Very Useful Fact

Believing more things can't make you any less inconsistent.

Lemma (monotonicity of inconsistency)

For all pdgs $\mathcal{m}$, $\mathcal{m}'$, *and all* $\gamma > 0$,

1. $\langle\!\langle \mathcal{m} \sqcup \mathcal{m}' \rangle\!\rangle_\gamma \geq \langle\!\langle \mathcal{m} \rangle\!\rangle_\gamma$.
2. *If* $\mathcal{m}$ *and* $\mathcal{m}'$ *have respective confidence vectors* β *and* β'*, and* $\beta \succeq \beta'$ *(that is,* $\beta_L \geq \beta'_L$ *for all* $L \in \mathcal{E}$*), then* $\langle\!\langle \mathcal{m} \rangle\!\rangle_\gamma \geq \langle\!\langle \mathcal{m}' \rangle\!\rangle_\gamma$.

Visual Proof: The Varitaional Bound

Visual Proof: The Varitaional Bound

$$\left\langle\!\!\left\langle \begin{tikzcd} {} \arrow[r, "p"] & Z \arrow[r, bend left, "d"] & X \arrow[l, bend left, "e!"] & {} \arrow[l, two heads, "x"'] \end{tikzcd} \right\rangle\!\!\right\rangle = -\,\mathrm{ELBO}_{p,e,d}(x).$$

Visual Proof: The Varitaional Bound

$$-\log \Pr_{p,d}(X{=}x) \;=\; \left\langle\!\!\left\langle \xrightarrow{p} \boxed{Z} \xrightarrow{d} \boxed{X} \overset{x}{\twoheadleftarrow} \right\rangle\!\!\right\rangle \qquad \left\langle\!\!\left\langle \xrightarrow{p} \boxed{Z} \underset{e!}{\overset{d}{\rightleftarrows}} \boxed{X} \overset{x}{\twoheadleftarrow} \right\rangle\!\!\right\rangle \;=\; -\,\mathrm{ELBO}_{p,e,d}(x).$$

Visual Proof: The Varitaional Bound

There's no way you'll have the time for this.

$$-\log \Pr_{p,d}(X{=}x) \;=\; \left\langle\!\!\left\langle \xrightarrow{p} \boxed{Z} \xrightarrow{d} \boxed{X} \overset{x}{\twoheadleftarrow} \right\rangle\!\!\right\rangle \leq \left\langle\!\!\left\langle \begin{tikzcd} {} \arrow[r, "p"] & \boxed{Z} \arrow[r, bend left, "d"] & \boxed{X} \arrow[l, bend left, "e!"] & {} \arrow[l, two heads, "x"'] \end{tikzcd} \right\rangle\!\!\right\rangle \;=\; -\mathrm{ELBO}_{p,e,d}(x).$$

Divergences and Inconsistency

Lemma (Divergences and PDGs)

The PDG divergence $\boldsymbol{D}^{\mathrm{PDG}}_{(r,s)}(p,q)$*, the inconsistency of a PDG containing* $p(X)$ *with confidence* r *and* $q(X)$ *with confidence* s*, is given by*

$$\boldsymbol{D}^{\mathrm{PDG}}_{(r,s)}(p,q) := \left\langle\!\!\left\langle \xrightarrow[(\beta:r)]{p} \boxed{X} \xleftarrow[(\beta:s)]{q} \right\rangle\!\!\right\rangle = -(r+s)\log\sum_x \Big(p(x)^r q(x)^s\Big)^{\frac{1}{r+s}}.$$

Visual Proof: Data Processing Inequality

$$\boldsymbol{D}\Big(p \,\Big\|\, q\Big) \geq \boldsymbol{D}\Big(f \circ p \,\Big\|\, f \circ q\Big)$$

$$\left\langle\!\!\left\langle \xrightarrow{\;p\;} \boxed{X} \xleftarrow{\;q\;} \right\rangle\!\!\right\rangle$$

$$\left\langle\!\!\left\langle \xrightarrow{f\circ p} \boxed{X} \xleftarrow{f\circ q} \right\rangle\!\!\right\rangle$$

Visual Proof: Data Processing Inequality

$$\boldsymbol{D}\Big(p \;\Big\|\; q\Big) \geq \boldsymbol{D}\Big(f \circ p \;\Big\|\; f \circ q\Big)$$

$$\left\langle\!\!\left\langle \xrightarrow{\;p\;} \boxed{X} \xleftarrow{\;q\;} \right\rangle\!\!\right\rangle = \left\langle\!\!\left\langle \begin{array}{c} \xrightarrow{\;p\;} \boxed{X} \xleftarrow{\;q\;} \\ {\scriptstyle f!}\big\downarrow \\ \boxed{Y} \end{array} \right\rangle\!\!\right\rangle$$

$$\left\langle\!\!\left\langle \xrightarrow{\;f \circ p\;} \boxed{X} \xleftarrow{\;f \circ q\;} \right\rangle\!\!\right\rangle$$

Visual Proof: Data Processing Inequality

$$\boldsymbol{D}\Big(p \,\Big\|\, q\Big) \geq \boldsymbol{D}\Big(f \circ p \,\Big\|\, f \circ q\Big)$$

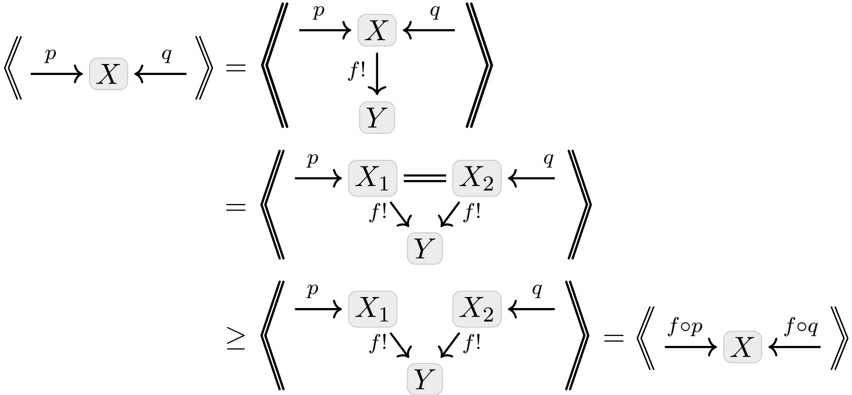
$$\left\langle\!\!\left\langle \xrightarrow{p} \boxed{X} \xleftarrow{q} \right\rangle\!\!\right\rangle = \left\langle\!\!\left\langle \begin{tikzcd} {} \arrow[r, "p"] & X \arrow[d, "f!"'] & {} \arrow[l, "q"'] \\ & Y & \end{tikzcd} \right\rangle\!\!\right\rangle$$

$$= \left\langle\!\!\left\langle \begin{tikzcd} {} \arrow[r, "p"] & X_1 \arrow[rr, equal] \arrow[dr, "f!"'] & & X_2 \arrow[dl, "f!"] & {} \arrow[l, "q"'] \\ & & Y & & \end{tikzcd} \right\rangle\!\!\right\rangle$$

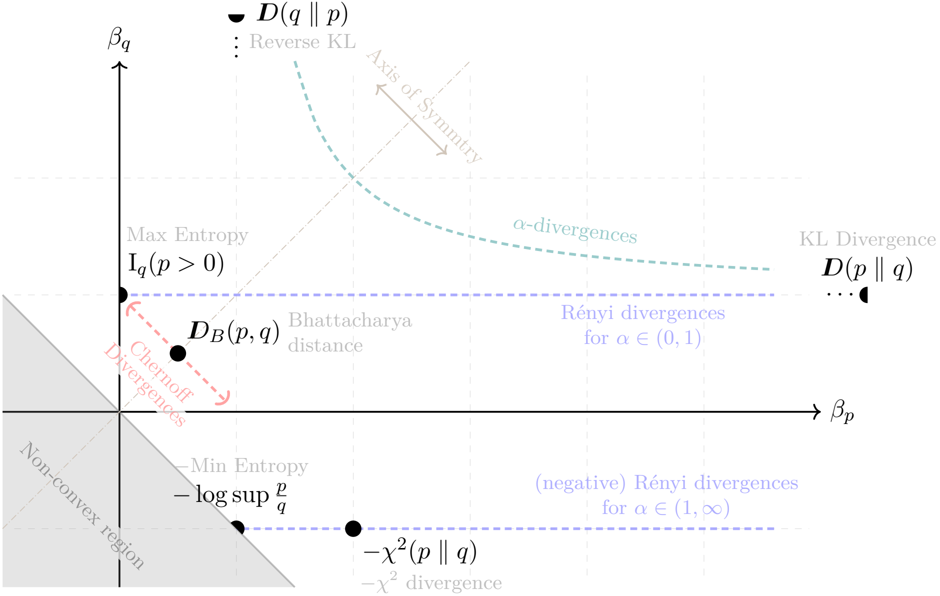
$$\left\langle\!\!\left\langle \xrightarrow{f \circ p} \boxed{X} \xleftarrow{f \circ q} \right\rangle\!\!\right\rangle$$

Visual Proof: Data Processing Inequality

$$\boldsymbol{D}\Big(p \,\Big\|\, q\Big) \geq \boldsymbol{D}\Big(f \circ p \,\Big\|\, f \circ q\Big)$$



Divergences as Inconsistencies



Outline for Section 7

$\langle$ Schema Picture $\rangle$
$\langle$ Add Universal Relation Theorem $\rangle$

Outline for Section 8

Open Problems and Future Work

$\langle$ INCOMPLETE $\rangle$

- - Trace Semantics
  - Composition

Summary

$\langle$ `update with second half` $\rangle$

PDGs...

- capture inconsistency, including conflicting information from multiple sources with varying reliability.

Summary

$\langle$ `update with second half` $\rangle$

PDGs...

- capture inconsistency, including conflicting information from multiple sources with varying reliability.
- are especially modular; to combine info from two sources, simply take a PDG union. This incorporates new data (edge cpds) and concepts (nodes) without affecting previous information.

Summary

$\langle$ `update with second half` $\rangle$

PDGs...

- capture inconsistency, including conflicting information from multiple sources with varying reliability.
- are especially modular; to combine info from two sources, simply take a PDG union. This incorporates new data (edge cpds) and concepts (nodes) without affecting previous information.
- cleanly separate quantitative info (the cpds) from qualitative info (the edges), with variable confidence in both (the weights β and α). This is captured by terms *Inc* and *IDef* in our scoring function.

Summary

⟨ `update with second half` ⟩

PDGs...

- capture inconsistency, including conflicting information from multiple sources with varying reliability.
- are especially modular; to combine info from two sources, simply take a PDG union. This incorporates new data (edge cpds) and concepts (nodes) without affecting previous information.
- cleanly separate quantitative info (the cpds) from qualitative info (the edges), with variable confidence in both (the weights β and α). This is captured by terms *Inc* and *IDef* in our scoring function.
- have (several) natural semantics; one of them allows us to pick out a unique distribution. Using this distrbution, PDGs can capture BNs and factor graphs.

Summary

⟨ update with second half ⟩

PDGs…

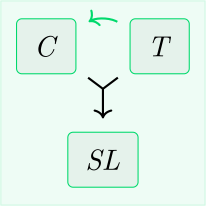
- capture inconsistency, including conflicting information from multiple sources with varying reliability.
- are especially modular; to combine info from two sources, simply take a PDG union. This incorporates new data (edge cpds) and concepts (nodes) without affecting previous information.
- cleanly separate quantitative info (the cpds) from qualitative info (the edges), with variable confidence in both (the weights β and α). This is captured by terms *Inc* and *IDef* in our scoring function.
- have (several) natural semantics; one of them allows us to pick out a unique distribution. Using this distrbution, PDGs can capture BNs and factor graphs.

But there is much more to be done!

$\langle$ `return to initial slide, but with more conflicts` $\rangle$

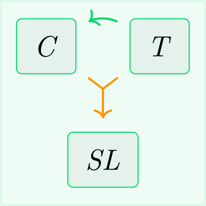
Outline for Section 9

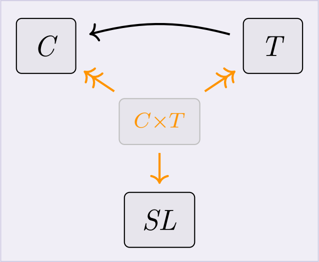
Hyper-graphs? Or merely graphs?



main definition

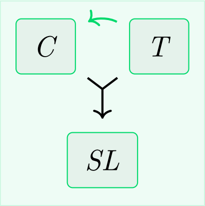
Hyper-graphs? Or merely graphs?

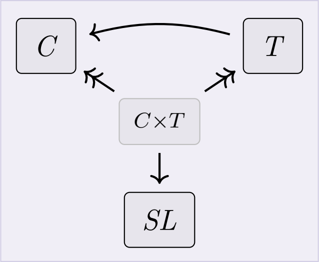




main definition

Hyper-graphs? Or merely graphs?

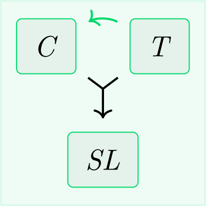


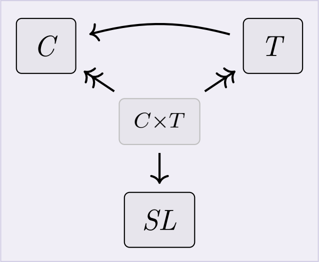


- This widget expands state space, but graphs are simpler.

main definition

Hyper-graphs? Or merely graphs?



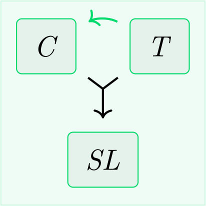


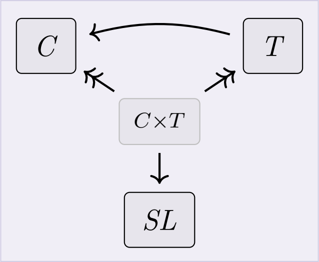
- This widget expands state space, but graphs are simpler.
- There is a natural correspondence

joint distributions $\leftrightarrows$ expanded joint distributions satisfying coherence constraints

main definition

Hyper-graphs? Or merely graphs?





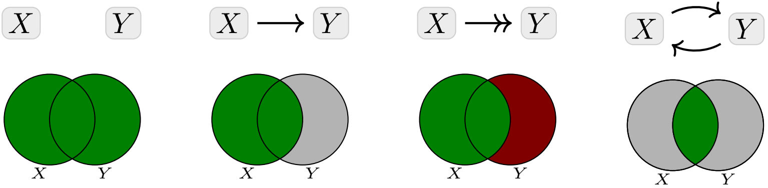
- This widget expands state space, but graphs are simpler.
- There is a natural correspondence

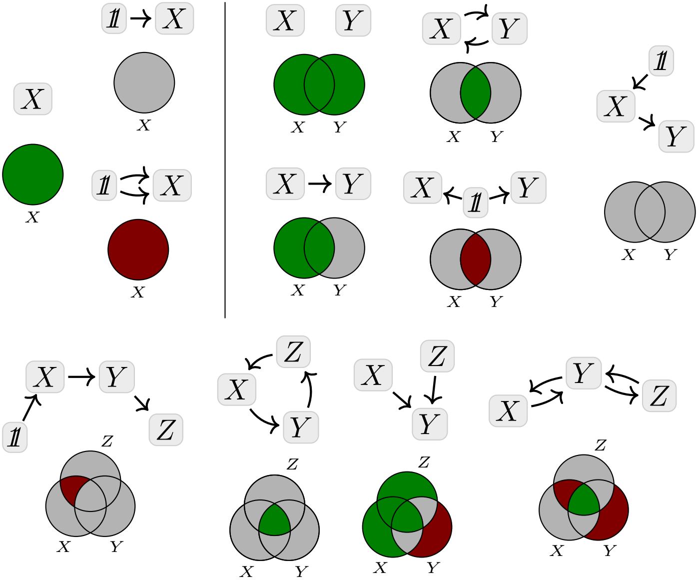
joint distributions $\leftrightarrows$ expanded joint distributions satisfying coherence constraints

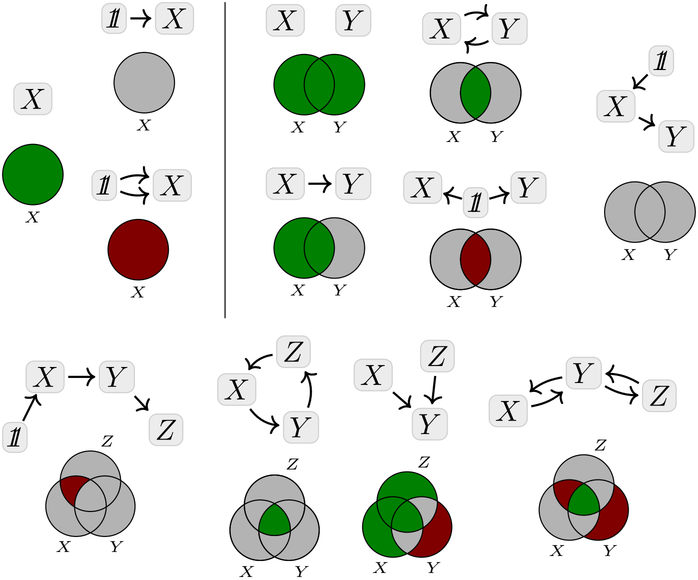
(working directly with hypergraphs is also possible)

main definition

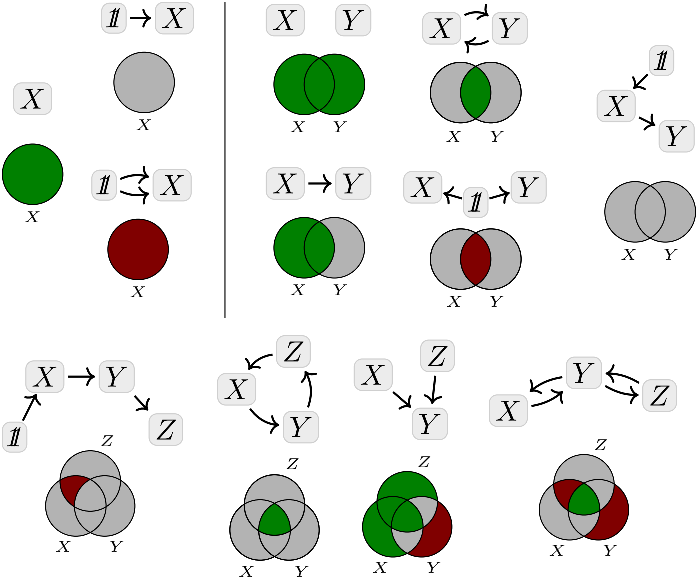
Illustrations of *IDef*

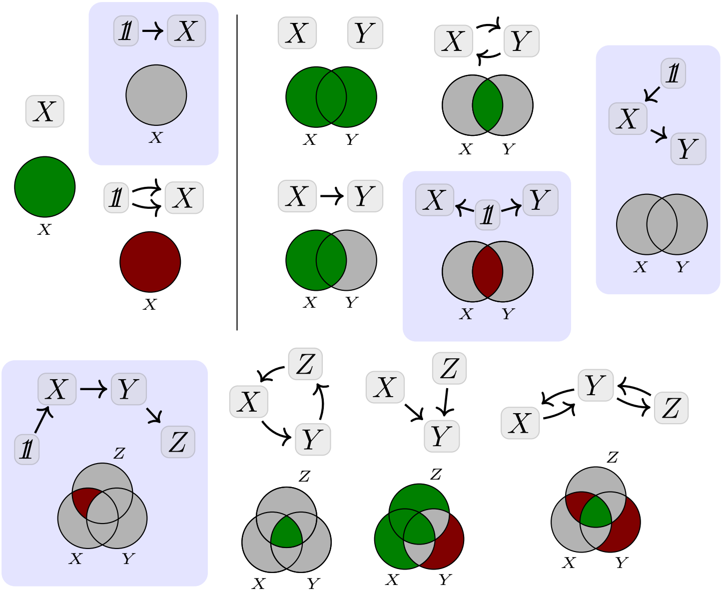



$\mathbb{1}$
X
Y
Z



X
1 → X
X
1 → X
X
X Y
X Y
X → Y
X Y
X Y
X Y
X ← 1 → Y
X Y
1
X
Y
X Y
X → Y
1
Z
Z
X
Y
Z
X Y
X
Z
Y
Z
X Y
Y
X
Z
Z
X Y





X
1 → X
X
1 ⇉ X
X
X Y
X Y
X ⇄ Y
X Y
X → Y
X Y
X ← 1 → Y
X Y
1 → X → Y
X Y
1 → X → Y → Z
Z
X Y
Z
X
Y
Z
X Y
X
Z
Y
Z
X Y
X
Y
Z
Z
X Y