Probabilistic Dependency Graphs and Inconsistency

How to model, measure, and mitigate internal conflict

Oliver Richardson

Cornell University
Department of Computer Science

September 2021

Outline for Section 1

The standard way of modeling an agent with uncertainty:

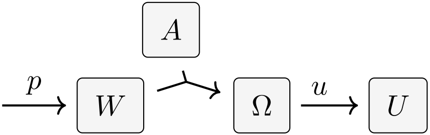
The standard way of modeling an agent with uncertainty:

- a probability distribution $p : \Delta W$ over worlds W,

$$\xrightarrow{p} \boxed{W}$$

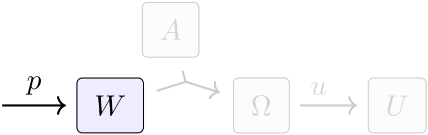
The standard way of modeling an agent with uncertainty:

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



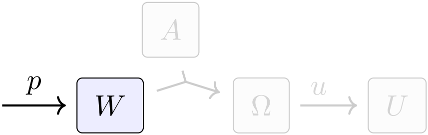
The standard way of modeling an agent with uncertainty:

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



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.

Why Inconsistency?

A man with a watch knows what time it is;
a man with two watches is never sure.
(Segal's Law)

Why Inconsistency?

> *A man with a watch knows what time it is;*
> *a man with two watches is never sure.*
> *(Segal's Law)*

Why **model** inconsistency?

- Sometimes people have inconsistent beliefs;

Why Inconsistency?

A man with a watch knows what time it is;
a man with two watches is never sure.
(Segal's Law)

Why **model** inconsistency?

- Sometimes people have inconsistent beliefs;
  - also want to understand the process of resolving it.

Why Inconsistency?

A man with a watch knows what time it is;
a man with two watches is never sure.
(Segal's Law)

Why **model** inconsistency?

- Sometimes people have inconsistent beliefs;
  - also want to understand the process of resolving it.

Why **construct** an agent that can be inconsistent?

Why Inconsistency?

> *A man with a watch knows what time it is;*
> *a man with two watches is never sure.*
> *(Segal's Law)*

Why **model** inconsistency?

- Sometimes people have inconsistent beliefs;
  - also want to understand the process of resolving it.

Why **construct** an agent that can be inconsistent?

- Perfect consistency can be (needlessly) expensive.

Why Inconsistency?

> *A man with a watch knows what time it is;*
> *a man with two watches is never sure.*
> *(Segal's Law)*

Why **model** inconsistency?

- Sometimes people have inconsistent beliefs;
  - also want to understand the process of resolving it.

Why **construct** an agent that can be inconsistent?

- Perfect consistency can be (needlessly) expensive.
- Useful for identifying big problems.

Why Inconsistency?

A man with a watch knows what time it is;
a man with two watches is never sure.
(Segal's Law)

Why **model** inconsistency?

- Sometimes people have inconsistent beliefs;
  - also want to understand the process of resolving it.

Why **construct** an agent that can be inconsistent?

- Perfect consistency can be (needlessly) expensive.
- Useful for identifying big problems.
  - (assertions, check-sums, paradoxes)

Why Inconsistency?

A man with a watch knows what time it is;
a man with two watches is never sure.
(Segal's Law)

Why **model** inconsistency?

- Sometimes people have inconsistent beliefs;
  - also want to understand the process of resolving it.

Why **construct** an agent that can be inconsistent?

- Perfect consistency can be (needlessly) expensive.
- Useful for identifying big problems.
  - (assertions, check-sums, paradoxes)

Freedom from prefect consistency is valuable, but demands the ability to recognize and adress internal conflict.

Yet Another Probabilistic Graphical Model

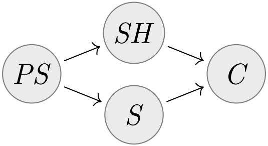
Probabilistic Dependency Graphs (PDGs),
a new class of graphical model designed to model inconsistent beliefs.

Yet Another Probabilistic Graphical Model

Probabilistic Dependency Graphs (PDGs),
a new class of graphical model designed to model inconsistent beliefs.

In doing so, we get much more . . .

Two aspects of Bayesian Networks (BNs)



Variables:

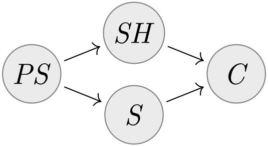
PS	Parents Smoke?
S	You smoke?
SH	Second-hand Smoke?
C	Get Cancer?

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



Variables:

PS	Parents Smoke?
S	You smoke?
SH	Second-hand Smoke?
C	Get Cancer?

Two aspects of Bayesian Networks (BNs)

Qualitative BN, $\mathcal{G}$

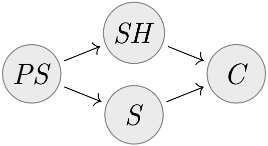
an independence relation on variables

- $X \perp\!\!\!\perp_{\mathcal{G}} Y \mid \mathbf{Pa}(X)$, 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 $\perp\!\!\!\perp_{\mathcal{G}}$.



Variables:

PS	Parents Smoke?
S	You smoke?
SH	Second-hand Smoke?
C	Get Cancer?

Outline for Section 2

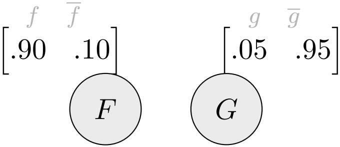
Simple Example: Floomps and Guns

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).

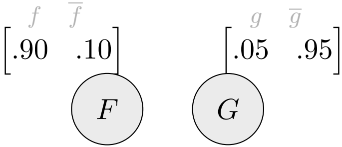
BN



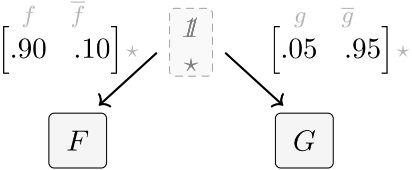
Simple Example: Floomps and Guns

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

BN



PDG

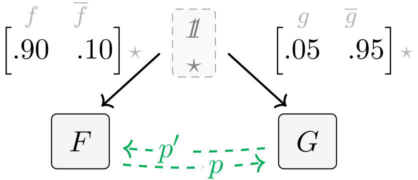


- The cpds of a PDG are attached to edges, not nodes.

Simple Example: Floomps and Guns

BN

PDG

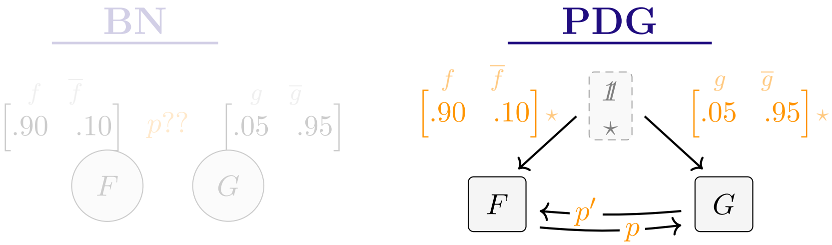


- The cpds of a PDG are attached to edges, not nodes.
- PDGs can incorporate arbitrary new probabilistic information.

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

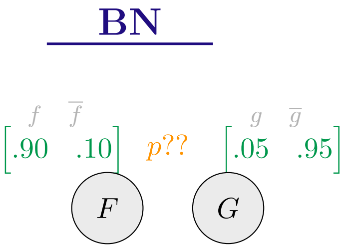
$$p(G\,|\,F) = \begin{array}{cc} \begin{array}{cc} g & \bar{g} \end{array} & \\ \begin{bmatrix} .92 & .08 \\ .08 & .92 \end{bmatrix} & \begin{array}{c} f \\ \bar{f} \end{array} \end{array} = \left(\, p'(F\,|\,G)\,\right)^{\mathsf{T}}$$

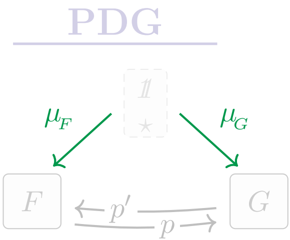
Simple Example: Floomps and Guns



- 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

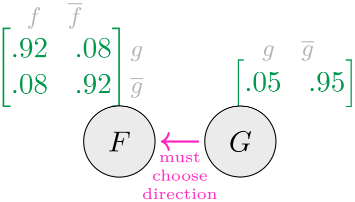




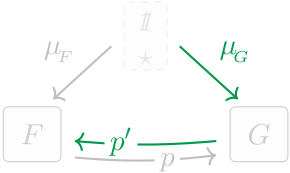
- The cpds of a PDG are attached to edges, not nodes.
- PDGs can incorporate arbitrary new probabilistic information.
- PDGs can be inconsistent,
  - ... but BNs must resolve inconsistency first,
    which may break symmetry and irrecoverably lose information.

Simple Example: Floomps and Guns

BN



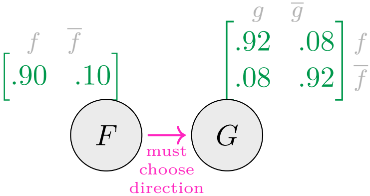
PDG



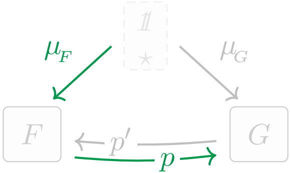
- The cpds of a PDG are attached to edges, not nodes.
- PDGs can incorporate arbitrary new probabilistic information.
- PDGs can be inconsistent,
  - ...but BNs must resolve inconsistency first,
    which may break symmetry and irrecoverably lose information.

Simple Example: Floomps and Guns

BN

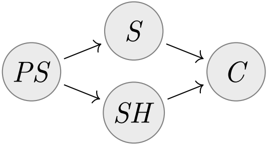


PDG



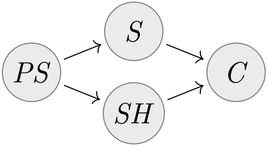
- The cpds of a PDG are attached to edges, not nodes.
- PDGs can incorporate arbitrary new probabilistic information.
- PDGs can be inconsistent,
  - ... but BNs must resolve inconsistency first, which may break symmetry and irrecoverably lose information.

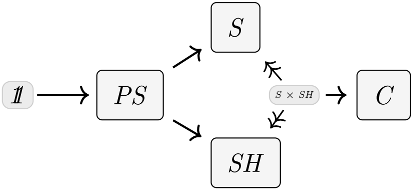
Bayesian Networks as PDGs



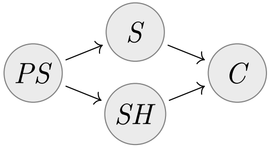
▸ Hypergraphs!

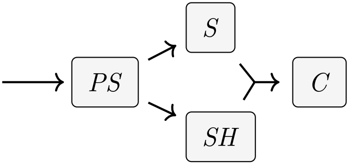
Bayesian Networks as PDGs



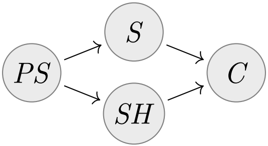


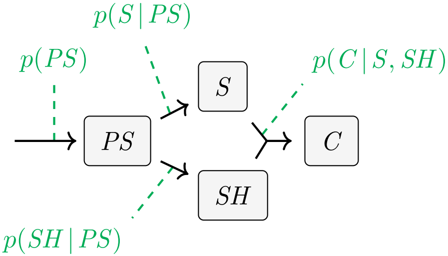
Bayesian Networks as PDGs





Bayesian Networks as PDGs

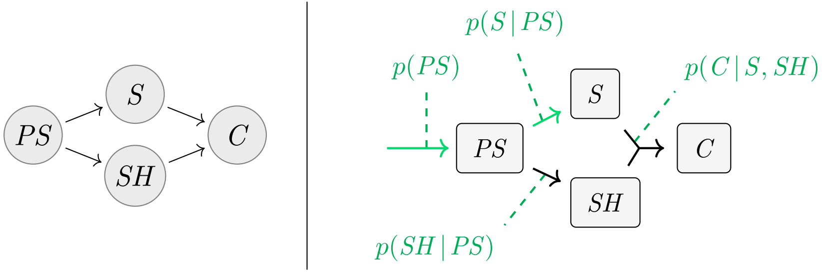




In contrast with BNs:

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

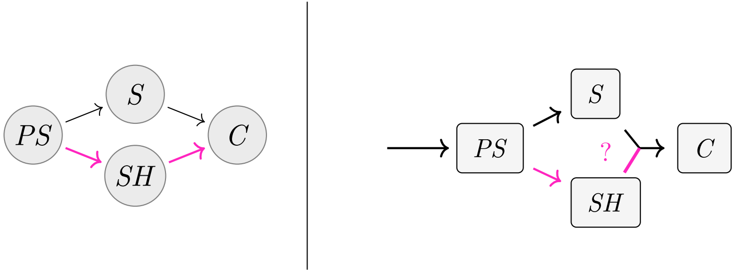
Bayesian Networks as PDGs



In contrast with BNs:

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

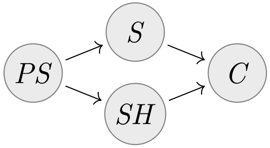
Bayesian Networks as PDGs

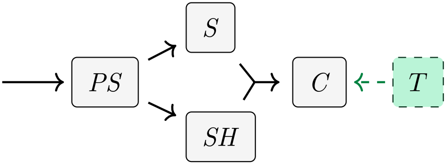


In contrast with BNs:

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

Bayesian Networks as PDGs

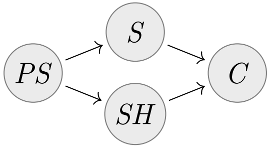


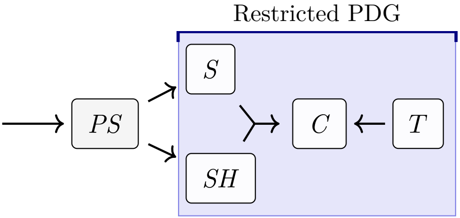


In contrast with BNs:

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

Bayesian Networks as PDGs

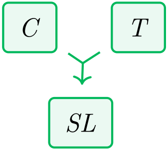




In contrast with BNs:

- 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.

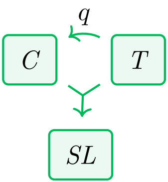
Combining PDGs



Grok wants to be supreme leader (SL).

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

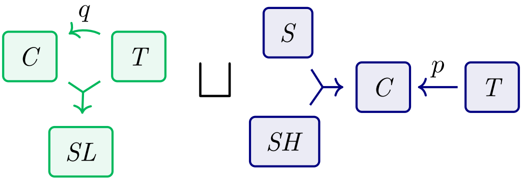
Combining PDGs



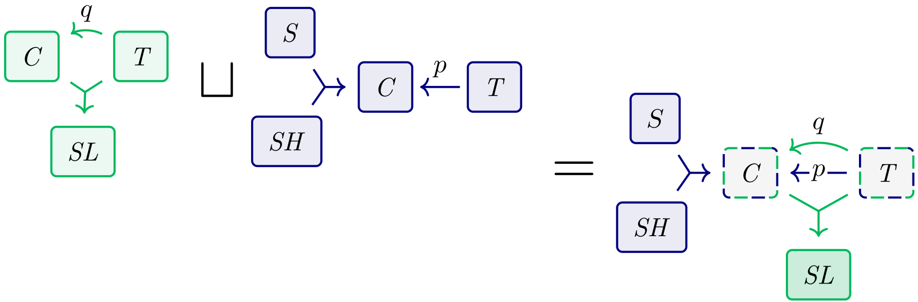
Grok wants to be supreme leader (SL).

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

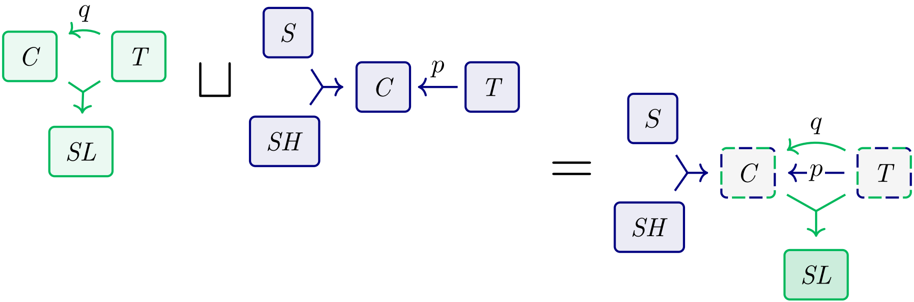
Combining PDGs



Combining PDGs

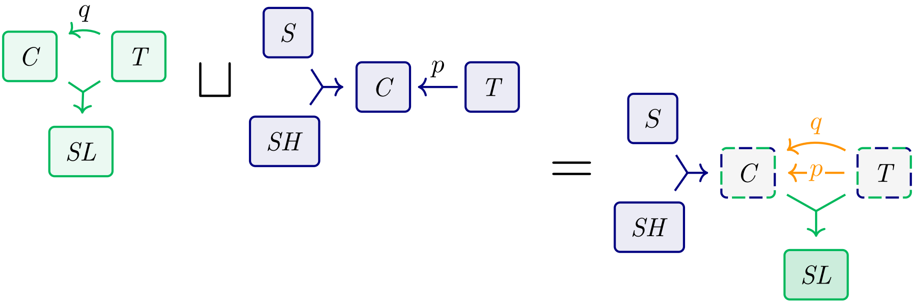


Combining PDGs



- 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 3

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 $\mathcal{m} = (\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p}, \alpha, \beta)$, 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 $\mathcal{M} = (\mathcal{N}, \mathcal{E}, \mathcal{V}, \mathbf{p}, \alpha, \beta)$, 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)

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

(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:

$\mathbf{p}_L$ 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 $\mathbf{p}_L$.

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 $\mathbf{p}_L$.

Outline for Section 4

Semantics of PDGs

How to stitch cpds together?

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

The set of joint distributions consistent with $\mathcal{m}$;

Semantics of PDGs

How to stitch cpds together?

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

The set of joint distributions consistent with $\mathcal{m}$;

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

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

The Scoring Function

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

The Scoring Function

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

Intuition: Measure μ's violation of $\mathcal{M}$'s cpds.

The Scoring Function

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

Definition (Inc)

The *incompatibility* of a joint distribution μ with $\mathcal{m}$ is given by

$$\mathit{Inc}_{\mathcal{m}}(\mu) := \sum_{X \xrightarrow{L} Y} \boldsymbol{D}(\mu_{Y|X} \parallel \mathbf{p}_{L})$$

The Scoring Function

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

Definition (Inc)

The *incompatibility* of a joint distribution μ with $\mathcal{m}$ is given by

$$\mathit{Inc}_{\mathcal{m}}(\mu) := \sum_{X \xrightarrow{L} Y} \beta_L\ \boldsymbol{D}(\mu_{Y|X} \parallel \mathbf{p}_L)$$

The Scoring Function

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

Definition (*Inc*)

The *incompatibility* of a joint distribution μ with $\mathcal{M}$ is given by

$$\mathit{Inc}_{\mathcal{M}}(\mu) := \sum_{X \xrightarrow{L} Y} \beta_L\, \boldsymbol{D}(\mu_{Y|X} \,\|\, \mathbf{p}_L)$$

$$\boldsymbol{D}(\mu \,\|\, \nu) = \sum_{w \in \operatorname{Supp}\mu} \mu(w) \log \frac{\mu(w)}{\nu(w)}$$

the relative entropy (KL Divergence) from ν to μ.

The Scoring Function

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

Intuition: each edge $X \xrightarrow{L} Y$ indicates that Y is determined (perhaps noisily) by X alone.

The Scoring Function

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

Intuition: each edge $X \xrightarrow{L} Y$ indicates that Y is determined (perhaps noisily) by X alone.

So a μ with uncertainty in Y after X is known (beyond pure noise) is qualitatively worse.

The Scoring Function

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

Intuition: each edge $X \xrightarrow{L} Y$ indicates that Y is determined (perhaps noisily) by X alone.

So a μ with $\overbrace{\text{uncertainty in } Y \text{ after } X \text{ is known}}^{\textit{measured by } \mathrm{H}(Y \mid X)}$ (beyond $\underbrace{\text{pure noise}}_{\mathrm{H}(\mu)}$) is qualitatively worse.

The Scoring Function

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

Definition (IDef)

The *information deficiency* of a distribution μ with respect to $\mathcal{m}$ is

$$\mathit{IDef}_{\mathcal{m}}(\mu) := \sum_{X \xrightarrow{L} Y} \alpha_L \, \mathrm{H}_\mu(Y \,|\, X) - \mathrm{H}(\mu).$$

The Scoring Function

$$[\![\mathcal{m}]\!]_\gamma(\mu) := Inc_{\mathcal{m}}(\mu) + \gamma \; IDef_{\mathcal{m}}(\mu)$$

Definition (*IDef*)

The *information deficiency* of a distribution μ with respect to $\mathcal{m}$ is

$$IDef_{\mathcal{m}}(\mu) := \sum_{X \xrightarrow{L} Y} \alpha_L \, \mathrm{H}_\mu(Y \,|\, X) - \underbrace{\mathrm{H}(\mu)}_{\text{(a) \# bits needed to determine all variables}}.$$

The Scoring Function

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

Definition (*IDef*)

The *information deficiency* of a distribution μ with respect to $\mathcal{m}$ is

$$\mathit{IDef}_{\mathcal{m}}(\mu) := \overbrace{\sum_{X \xrightarrow{L} Y} \alpha_L \,\mathrm{H}_\mu(Y \,|\, X)}^{\text{(b) \# bits required to separately determine each target, knowing the source}} - \underbrace{\mathrm{H}(\mu)}_{\text{(a) \# bits needed to determine all variables}}.$$

The Scoring Function

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

Definition (*IDef*)

The *information deficiency* of a distribution μ with respect to $\mathcal{m}$ is

$$\mathit{IDef}_{\mathcal{m}}(\mu) := \overbrace{\sum_{X \xrightarrow{L} Y} \alpha_L \,\mathrm{H}_\mu(Y \mid X)}^{\text{(b) \# bits required to separately determine each target, knowing the source}} - \underbrace{\mathrm{H}(\mu)}_{\text{(a) \# bits needed to determine all variables}}.$$

The Scoring Function

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

tradeoff parameter $\gamma \geq 0$

Definition (*Inc*)

The *incompatibility* of μ with $\mathcal{m}$:

$$\mathit{Inc}_{\mathcal{m}}(\mu) := \sum_{X \xrightarrow{L} Y} \beta_L\ \boldsymbol{D}(\mu_{Y|X} \parallel \mathbf{p}_L)$$

Definition (*IDef*)

The $\mathcal{m}$-*information deficiency* of μ:

$$\mathit{IDef}_{\mathcal{m}}(\mu) = \sum_{X \xrightarrow{L} Y} \alpha_L \mathrm{H}_\mu(Y \mid X) - \mathrm{H}(\mu)$$

The Scoring Function

- We are interested in the quantitative limit (small γ)

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

Definition (*Inc*)

The *incompatibility* of μ with $\mathcal{m}$:

$$\mathit{Inc}_{\mathcal{m}}(\mu) := \sum_{X \xrightarrow{L} Y} \beta_L\ \boldsymbol{D}(\mu_{Y|X} \parallel \mathbf{p}_L)$$

Definition (*IDef*)

The $\mathcal{m}$-*information deficiency* of μ:

$$\mathit{IDef}_{\mathcal{m}}(\mu) = \sum_{X \xrightarrow{L} Y} \alpha_L \mathrm{H}_\mu(Y \mid X) - \mathrm{H}(\mu)$$

The Optimal Distribution(s)

We have a scoring function $[\![m]\!]_\gamma : \Delta\mathcal{V}(m) \to \mathbb{R}$.

▸ Formal Statement + properties ▸ Relationships Between Semantics

The Optimal Distribution(s)

We have a scoring function $[\![\mathcal{m}]\!]_\gamma : \Delta\mathcal{V}(\mathcal{m}) \to \mathbb{R}$.
Let $[\![\mathcal{m}]\!]^*_\gamma$ be the set of best-scoring distributions.

▸ Formal Statement + properties ▸ Relationships Between Semantics

The Optimal Distribution(s)

We have a scoring function $[\![\mathcal{m}]\!]_\gamma : \Delta\mathcal{V}(\mathcal{m}) \to \mathbb{R}$.
Let $[\![\mathcal{m}]\!]^*_\gamma$ be the set of best-scoring distributions.

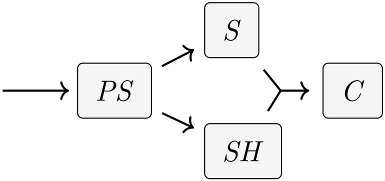
Proposition (uniqueness for small γ, informal)

As $\gamma \to 0$, there is a unique optimal distribution, which we call $[\![\mathcal{m}]\!]^*$.

▸ Formal Statement + properties ▸ Relationships Between Semantics

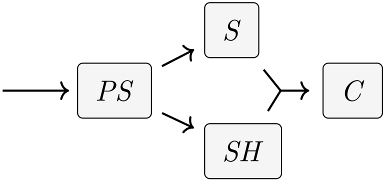
Outline for Section 5

Capturing Bayesian Networks



Capturing Bayesian Networks

For any β, let $\mathcal{m}_{\mathcal{B},\beta}$ be the PDG corresponding to $\mathcal{B}$, with $\alpha = \mathbf{1}$, and confidences β.



Capturing Bayesian Networks

For any β, let $\mathcal{m}_{\mathcal{B},\beta}$ be the PDG corresponding to $\mathcal{B}$, with $\alpha = \mathbf{1}$, and confidences β.

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,

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

▸ Corolary: BNs as Maximum Entropy Distributions

Capturing Bayesian Networks

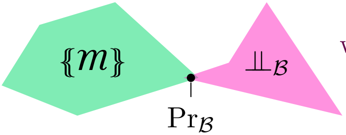
For any β, let $\mathcal{m}_{\mathcal{B},\beta}$ be the PDG corresponding to $\mathcal{B}$, with $\alpha = \mathbf{1}$, and confidences β.

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,

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

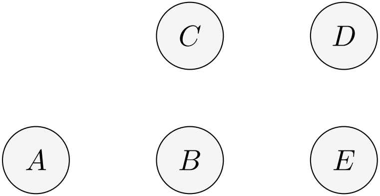
space of distributions consistent with $\mathcal{m}_{\mathcal{B}}$ (which minimize *Inc*)



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

▸ Corolary: BNs as Maximum Entropy Distributions

Factor Graphs

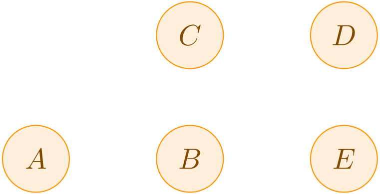


Definition

A *factor graph* Φ is

‣ Φ's standard scoring function: "variational free energy"

Factor Graphs

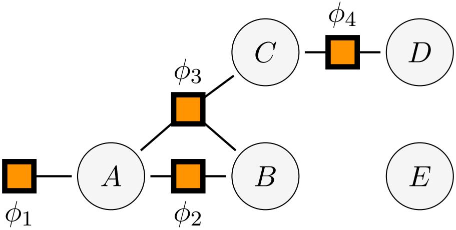


Definition

A *factor graph* Φ is a set of variables $\mathcal{X} = \{X_i\}$,

▸ Φ's standard scoring function: "variational free energy"

Factor Graphs

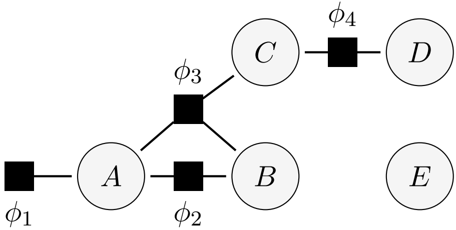


Definition

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

▸ Φ's standard scoring function: "variational free energy"

Factor Graphs



Definition

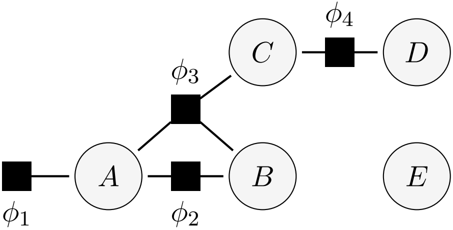
A *factor graph* Φ is a set of variables $\mathcal{X} = \{X_i\}$, and *factors* $\{\phi_J \colon \mathcal{V}(X_J) \to \mathbb{R}_{\geq 0}\}_{J \in \mathcal{J}}$, with $X_J \subseteq \mathcal{X}$; Φ defines a distribution

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

where Z_Φ is the normalization constant.

▸ Φ's standard scoring function: "variational free energy"

Factor Graphs



Definition

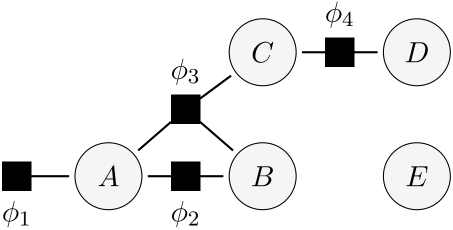
A *factor graph* Φ is a set of variables $\mathcal{X} = \{X_i\}$, and *factors* $\{\phi_J : \mathcal{V}(X_J) \to \mathbb{R}_{\geq 0}\}_{J \in \mathcal{J}}$, with $X_J \subseteq \mathcal{X}$; Φ defines a distribution

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

where Z_Φ is the normalization constant.

▸ Φ's standard scoring function: "variational free energy"

Factor Graphs



Definition

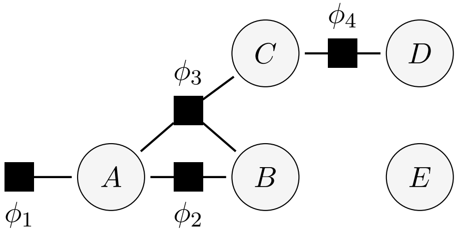
A *factor graph* Φ is a set of variables $\mathcal{X} = \{X_i\}$, and *factors* $\{\phi_J : \mathcal{V}(X_J) \to \mathbb{R}_{\geq 0}\}_{J \in \mathcal{J}}$, with $X_J \subseteq \mathcal{X}$; Φ defines a distribution

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

where Z_Φ is the normalization constant.

▸ Ψ's standard scoring function: "variational free energy"

Factor Graphs



Definition

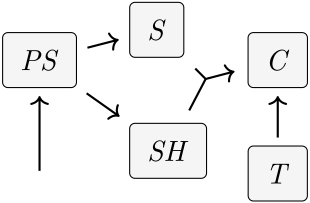
A weighted *factor graph* $\Psi = (\Phi, \theta)$ is a set of variables $\mathcal{X} = \{X_i\}$, *factors* $\{\phi_J : \mathcal{V}(X_J) \to \mathbb{R}_{\geq 0}\}_{J \in \mathcal{J}}$, and weights $(\theta_J)_{J \in \mathcal{J}}$ with $X_J \subseteq \mathcal{X}$; Ψ defines a distribution

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

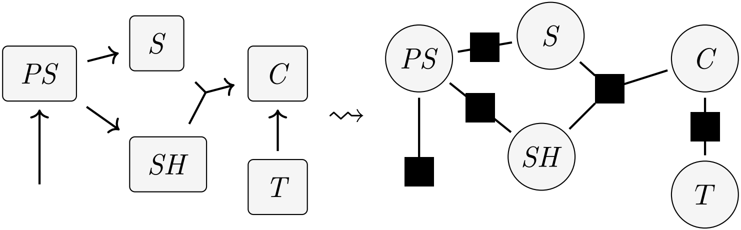
where Z_Ψ is the normalization constant.

▸ Ψ's standard scoring function: "variational free energy"

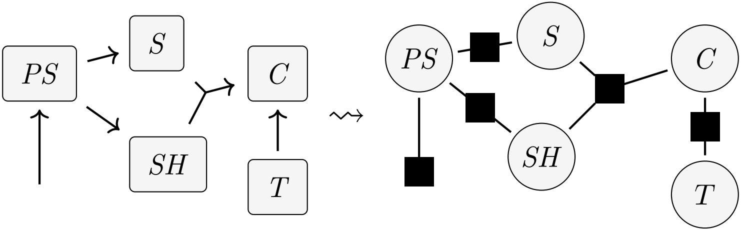
PDGs as Factor Graphs



PDGs as Factor Graphs

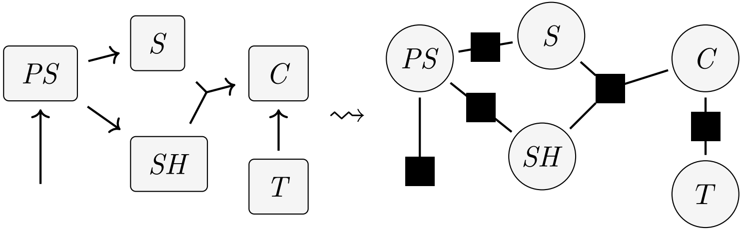


PDGs as Factor Graphs



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

PDGs as Factor Graphs



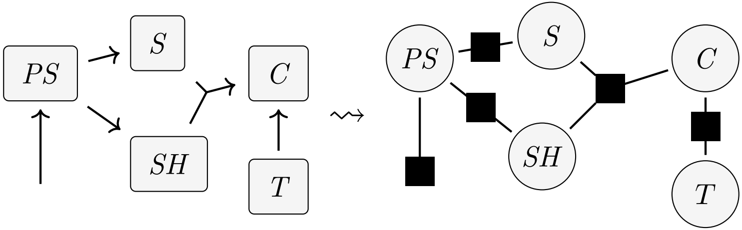
The cpds of a PDG are essentially factors. Are the semantics the same?

Theorem (Yes, for $\gamma = 1$)

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

Precise Theorem Statement

PDGs as Factor Graphs



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

Theorem (Yes, for $\gamma = 1$)

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

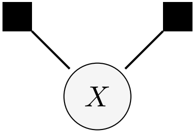
Theorem (generalization to weighted factor graphs)

Semantics match (for specific γ) if $\boldsymbol{\beta} \propto \boldsymbol{\alpha}$.

▸ Precise Theorem Statement

An Important Difference between PDGs and Factor Graphs

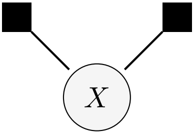
$$m := \begin{tikzcd} \mathbb{1} \arrow[d, bend right, "q"'] \arrow[d, bend left, "p"] \\ X \end{tikzcd}$$



$=: \Phi$

An Important Difference between PDGs and Factor Graphs

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

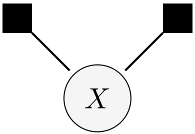


$=: \Phi$

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

An Important Difference between PDGs and Factor Graphs

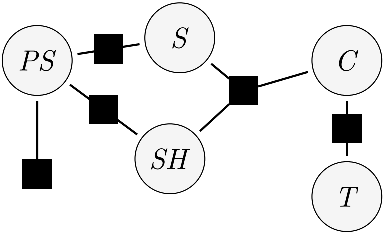
$$m := \begin{tikzcd} \boxed{\mathbb{1}} \arrow[d, bend right, "q"'] \arrow[d, bend left, "p"] \\ \boxed{X} \end{tikzcd}$$



$=: \Phi$

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

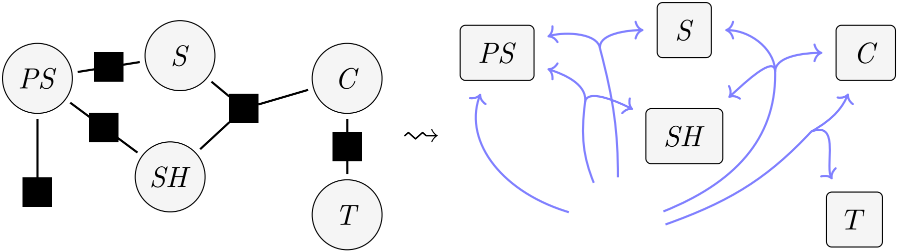
Factor Graphs as PDGs



$\rightsquigarrow$

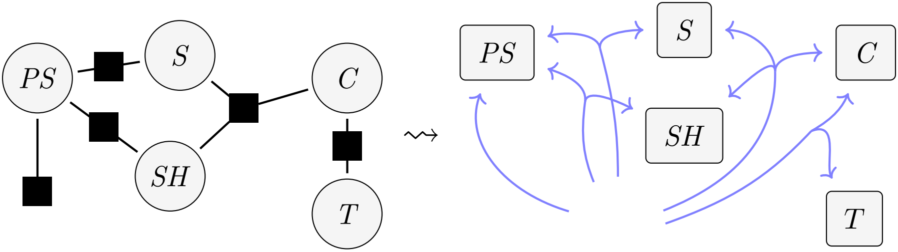
▸ Full Theorem

Factor Graphs as PDGs



Full Theorem

Factor Graphs as PDGs

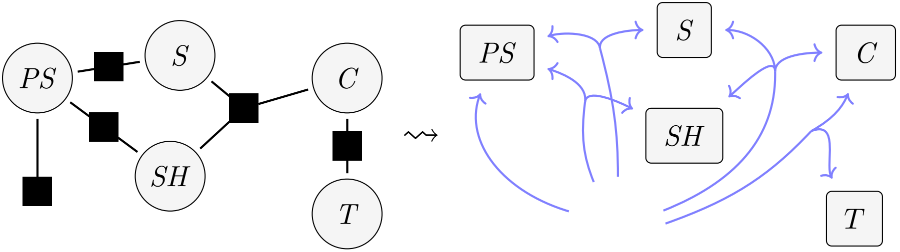


Theorem

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

▸ Full Theorem

Factor Graphs as PDGs



Theorem

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

Theorem

A simlar result holds for weighted factor graphs.

▸ Full Theorem

Outline for Section 6

1. Introduction
2. Modeling Examples
   - What are Floomps?
   - Smoking BN Manipulations
   - Union and Restriction
3. Syntax
4. Semantics
5. Capturing other Graphical Models
   - Bayesian Networks
   - Factor Graphs
6. Inference
7. Inconsistency as Loss
   - Motivation
   - Standard Metrics
   - Inconsistency and Statistical Divergences
   - Variational AutoEncoders
8. Other Aspects of PDGs
   - Category Theory
   - Databases
   - Other Projects Work

Inconsistency

Inconsistency: the optimal value of the scoring function.

$$\langle\!\langle m \rangle\!\rangle_\gamma := \inf_\mu \ [\![m]\!]_\gamma(\mu)$$

Inconsistency

Inconsistency: the optimal value of the scoring function.

$$\langle\!\langle m \rangle\!\rangle_\gamma := \inf_\mu \; [\![m]\!]_\gamma(\mu)$$

Nice property for minimization:

- $\langle\!\langle m \rangle\!\rangle$ is strictly convex and smooth in cpds (on the interior)

Inference via Inconsistency Reduction

Identify the event $Y{=}y$ with the cpd $\xrightarrow{\delta_y} Y$.

Inference via Inconsistency Reduction

Identify the event $Y{=}y$ with the cpd $\xrightarrow{\delta_y} Y$.

Conditioning as inconsistency resolution.

To condition on an event $(Y{=}y)$, simply add it to the PDG. Then the new best distribution is the old one, conditioned on $(Y{=}y)$. That is,

$$\left[\!\left[\mathcal{m} \sqcup (Y{=}y) \right]\!\right]^* = [\![\mathcal{m}]\!]^* \mid (Y{=}y).$$

Inference via Inconsistency Reduction

Identify the event $Y{=}y$ with the cpd $\xrightarrow{\delta_y} Y$.

Conditioning as inconsistency resolution.

To condition on an event $(Y{=}y)$, simply add it to the PDG. Then the new best distribution is the old one, conditioned on $(Y{=}y)$. That is,

$$\left[\!\left[\mathcal{m} \sqcup (Y{=}y) \right]\!\right]^* = [\![\mathcal{m}]\!]^* \mid (Y{=}y).$$

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

Inference via Inconsistency Reduction

Identify the event $Y{=}y$ with the cpd $\xrightarrow{\delta_y} Y$.

Conditioning as inconsistency resolution.
To condition on an event $(Y{=}y)$, simply add it to the PDG. Then the new best distribution is the old one, conditioned on $(Y{=}y)$. That is,

$$\Big[\!\!\Big[\mathcal{m} \sqcup (Y{=}y) \Big]\!\!\Big]^* = [\![\mathcal{m}]\!]^* \mid (Y{=}y).$$

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

- We can add $X \xrightarrow{p} Y$ to $\mathcal{m}$, to get $\mathcal{m} \sqcup p$.

Inference via Inconsistency Reduction

Identify the event $Y{=}y$ with the cpd $\xrightarrow{\delta_y} Y$.

Conditioning as inconsistency resolution.
To condition on an event $(Y{=}y)$, simply add it to the PDG. Then the new best distribution is the old one, conditioned on $(Y{=}y)$. That is,

$$\Big[\!\!\Big[\mathcal{m} \sqcup (Y{=}y) \Big]\!\!\Big]^* = [\![\mathcal{m}]\!]^* \mid (Y{=}y).$$

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

- We can add $X \xrightarrow{p} Y$ to $\mathcal{m}$, to get $\mathcal{m} \sqcup p$.
- The choice of cpd p that minimizes the inconsistency $\langle\!\langle \mathcal{m} \sqcup p \rangle\!\rangle$ is $[\![\mathcal{m}]\!]^*(Y \,|\, X)$,

Inference via Inconsistency Reduction

Identify the event $Y{=}y$ with the cpd $\xrightarrow{\delta_y} Y$.

Conditioning as inconsistency resolution.
To condition on an event $(Y{=}y)$, simply add it to the PDG. Then the new best distribution is the old one, conditioned on $(Y{=}y)$. That is,

$$\left[\!\left[\mathcal{m} \sqcup (Y{=}y) \right]\!\right]^* = [\![\mathcal{m}]\!]^* \mid (Y{=}y).$$

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

- We can add $X \xrightarrow{p} Y$ to $\mathcal{m}$, to get $\mathcal{m} \sqcup p$.
- The choice of cpd p that minimizes the inconsistency $\langle\!\langle \mathcal{m} \sqcup p \rangle\!\rangle$ is $[\![\mathcal{m}]\!]^*(Y \,|\, X)$,
  - so an inconsistency oracle yields fast inference by gradient descent.

Inference via Inconsistency Reduction

Identify the event $Y{=}y$ with the cpd $\xrightarrow{\delta_y} Y$.

Conditioning as inconsistency resolution.
To condition on an event $(Y{=}y)$, simply add it to the PDG. Then the new best distribution is the old one, conditioned on $(Y{=}y)$. That is,

$$\left[\!\!\left[\mathcal{m} \sqcup (Y{=}y)\right]\!\!\right]^* = [\![\mathcal{m}]\!]^* \mid (Y{=}y).$$

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

- We can add $X \xrightarrow{p} Y$ to $\mathcal{m}$, to get $\mathcal{m} \sqcup p$.
- The choice of cpd p that minimizes the inconsistency $\langle\!\langle \mathcal{m} \sqcup p \rangle\!\rangle$ is $[\![\mathcal{m}]\!]^*(Y \mid X)$,
  - so an inconsistency oracle yields fast inference by gradient descent.

(Theorem): Unfortunately,

1. Deciding if $\mathcal{m}$ is consistent is NP-hard.
2. Computing $\langle\!\langle \mathcal{m} \rangle\!\rangle_\gamma$ is #P-hard, for $\gamma > 0$.

Inference via Inconsistency Reduction

Identify the event $Y{=}y$ with the cpd $\xrightarrow{\delta_y} Y$.

Conditioning as inconsistency resolution.
To condition on an event $(Y{=}y)$, simply add it to the PDG. Then the new best distribution is the old one, conditioned on $(Y{=}y)$. That is,

$$\left[\!\left[\mathcal{m} \sqcup (Y{=}y) \right]\!\right]^* = [\![\mathcal{m}]\!]^* \mid (Y{=}y).$$

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

- We can add $X \xrightarrow{p} Y$ to $\mathcal{m}$, to get $\mathcal{m} \sqcup p$.
- The choice of cpd p that minimizes the inconsistency $\langle\!\langle \mathcal{m} \sqcup p \rangle\!\rangle$ is $[\![\mathcal{m}]\!]^*(Y \,|\, X)$,
  - so an inconsistency oracle yields fast inference by gradient descent.

(Theorem): Unfortunately,

1. Deciding if $\mathcal{m}$ is consistent is NP-hard.
2. Computing $\langle\!\langle \mathcal{m} \rangle\!\rangle_\gamma$ is #P-hard, for $\gamma > 0$.

. . . just like for BNs and Factor Graphs.

Outline for Section 7

1. Introduction
2. Modeling Examples
   - What are Floomps?
   - Smoking BN Manipulations
   - Union and Restriction
3. Syntax
4. Semantics
5. Capturing other Graphical Models
   - Bayesian Networks
   - Factor Graphs
6. Inference
7. **Inconsistency as Loss**
   - Motivation
   - Standard Metrics
   - Inconsistency and Statistical Divergences
   - Variational AutoEncoders
8. Other Aspects of PDGs
   - Category Theory
   - Databases
   - Other Projects Work

Inconsistency: The Universal Loss

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

Priors and Regularizers

Inconsistency: The Universal Loss

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

▸ Priors and Regularizers

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.

Priors and Regularizers

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.

Priors and Regularizers

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.
  - Model makes claims about reality.

▸ Priors and Regularizers

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.
  - Model makes claims about reality.
  - For instance: priors correspond to regularizers, but can be wrong.

Priors and Regularizers

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.
  - Model makes claims about reality.
  - For instance: priors correspond to regularizers, but can be wrong.

$$\text{Bayes Rule: posterior} \propto \text{likelihood} \cdot \textcolor{red}{\text{prior}}$$

▸ Priors and Regularizers

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.
  - Model makes claims about reality.
  - For instance: priors correspond to regularizers, but can be wrong.

$$\text{Bayes Rule: posterior} \propto \text{likelihood} \cdot \text{prior}$$
$$\log \text{posterior} = \log \text{likelihood} + \log \text{prior} + C$$

▸ Priors and Regularizers

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.
  - Model makes claims about reality.
  - For instance: priors correspond to regularizers, but can be wrong.

$$\text{Bayes Rule: } \text{posterior} \propto \text{likelihood} \cdot \text{prior}$$

$$\underset{\text{(new objective)}}{\log \text{posterior}} = \underset{\text{(old objective)}}{\log \text{likelihood}} + \underset{\text{(regularizer)}}{\log \text{prior}} + C$$

Priors and Regularizers

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.
  - Model makes claims about reality.
  - For instance: priors correspond to regularizers, but can be wrong.

Surprising Result

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

▸ Priors and Regularizers

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.
  - Model makes claims about reality.
  - For instance: priors correspond to regularizers, but can be wrong.

Surprising Result

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

Bonus

A visual language for reasoning about relationships between objectives.

▸ Priors and Regularizers

Surprise as Inconsistency

Consider a distribution $p(X)$.

Surprise as Inconsistency

Consider a distribution $p(X)$.
The surprise (information content) at seeing a sample x is:

$$\mathrm{I}_p(x) := \log \frac{1}{p(X{=}x)}.$$

Surprise as Inconsistency

Consider a distribution $p(X)$.
The surprise (information content) at seeing a sample x is:

$$\mathrm{I}_p(x) := \log \frac{1}{p(X\!=\!x)}.$$

Proposition

Surprise is the inconsistency of simultaneously believing p and $X = x$. That is,

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

Surprise as Inconsistency

Consider a distribution $p(X)$.
The surprise (information content) at seeing a sample x is:

$$\mathrm{I}_p(x) := \log \frac{1}{p(X\!=\!x)}.$$

Proposition

Surprise is the inconsistency of simultaneously believing p and $X = x$. That is,

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

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

Surprise as Inconsistency

Consider a distribution $p(X)$.
The surprise (information content) at seeing a sample x is:

$$\mathrm{I}_p(x) := \log \frac{1}{p(X\!=\!x)}.$$

Proposition

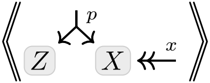
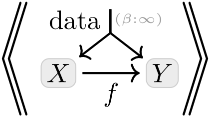
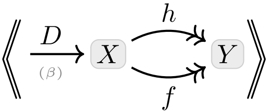
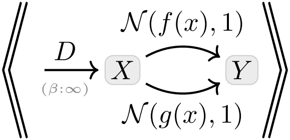
Surprise is the inconsistency of simultaneously believing p and $X = x$. That is,

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

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

For entire dataset

Big Table of Objectives

Objective	PDG	Equation
Marginal Information		$-\log p(X = x)$
Cross Entropy (Supervised)		$\dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\left[\log \frac{1}{f(y^i \mid x^i)}\right] - \mathrm{H}_{\text{data}}(Y \mid X)$
Accuracy		$-\beta \log \Big(\text{accuracy}_{f,D}(h)\Big)$
Square Loss		$\mathbb{E}_D \Big(f(X) - h(X)\Big)^2$

Priors and Regularizers

Inconsistency as a Divergence

You believe both $p(X)$ and $q(X)$.

$$\xrightarrow{p} \boxed{X} \xleftarrow{q}$$

Inconsistency as a Divergence

You believe both $p(X)$ and $q(X)$.
Your inconsistency: a divergence between p and q?

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

Inconsistency as a Divergence

You believe both $p(X)$ and $q(X)$.
Your inconsistency: a divergence between p and q?

$$\left\langle\!\!\left\langle \xrightarrow[(\beta:r)]{p} \boxed{X} \xleftarrow[(\beta:s)]{q} \right\rangle\!\!\right\rangle$$

Inconsistency as a Divergence

You believe both $p(X)$ and $q(X)$.
Your inconsistency: a divergence between p and q?

$$\text{Let } \boldsymbol{D}^{\mathrm{PDG}}_{(r,s)}(p,q) := \left\langle\!\!\left\langle \xrightarrow[(\beta:r)]{p} \boxed{X} \xleftarrow[(\beta:s)]{q} \right\rangle\!\!\right\rangle$$

Inconsistency as a Divergence

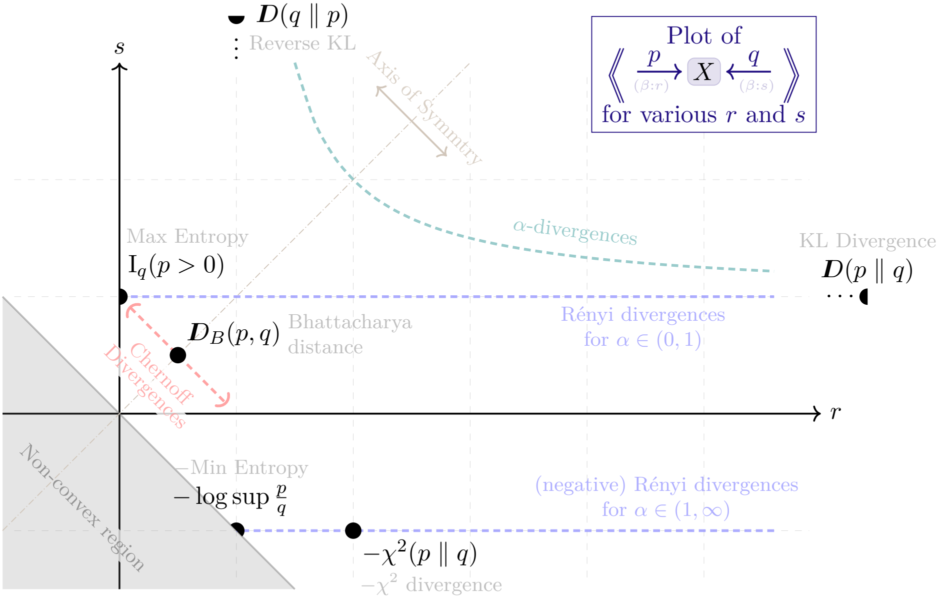
You believe both $p(X)$ and $q(X)$.
Your inconsistency: a divergence between p and q?

$$\text{Let } \boldsymbol{D}^{\mathrm{PDG}}_{(r,s)}(p,q) := \left\langle\!\!\left\langle \xrightarrow[(\beta:r)]{p} \boxed{X} \xleftarrow[(\beta:s)]{q} \right\rangle\!\!\right\rangle$$

Lemma (Closed Form)

$$\boldsymbol{D}^{\mathrm{PDG}}_{(r,s)}(p,q) = -(r+s)\log \sum_x \Big(p(x)^r q(x)^s \Big)^{\frac{1}{r+s}} .$$

Divergences as Inconsistencies



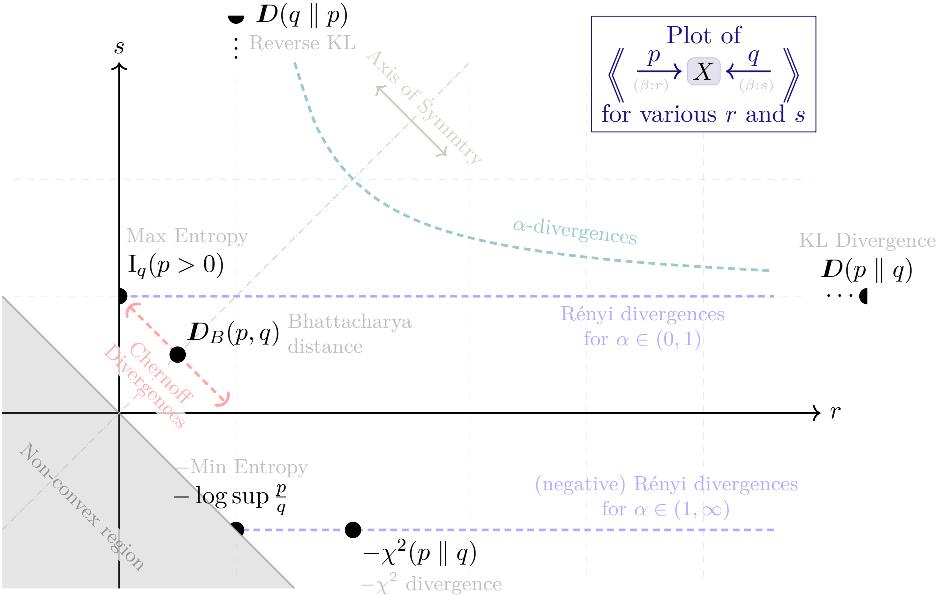
A Very Useful Fact

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

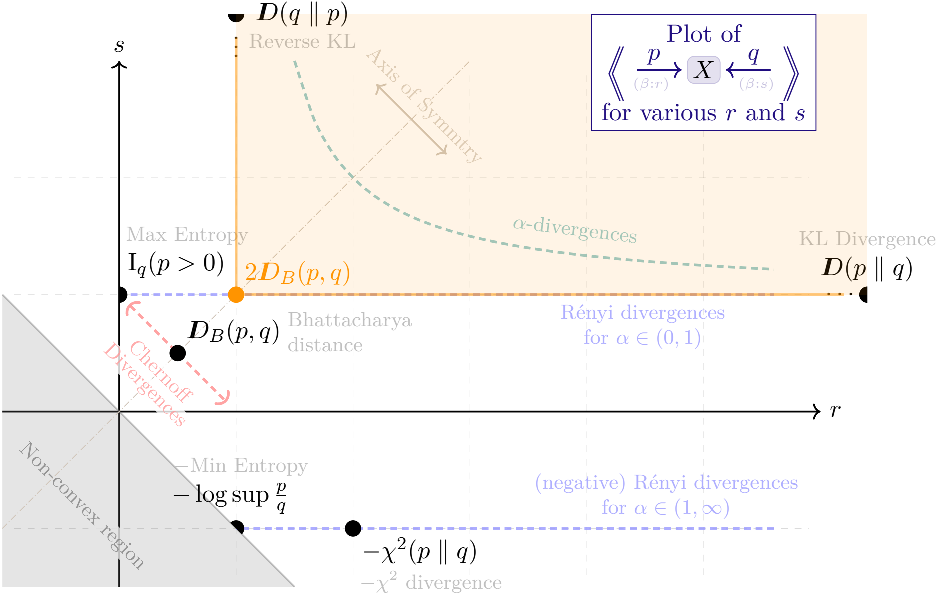
Lemma (monotonicity of inconsistency)

1. $\langle\!\langle \mathcal{m} \sqcup \mathcal{m}' \rangle\!\rangle \geq \langle\!\langle \mathcal{m} \rangle\!\rangle$.
2. *If* $\beta > \beta'$, *then* $\langle\!\langle \mathcal{m} \rangle\!\rangle \geq \langle\!\langle \mathcal{m}' \rangle\!\rangle$.

Divergences as Inconsistencies

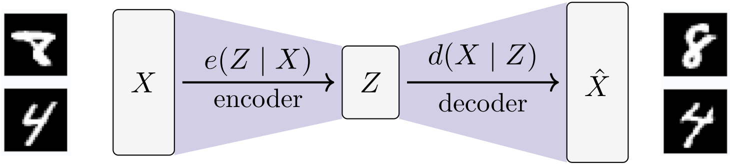


Divergences as Inconsistencies



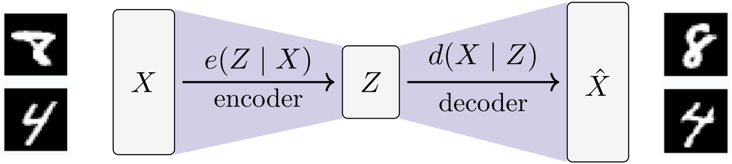
Variational Auto-Encoders, Take 1

- Structure consists of two neural networks (cpds):



Variational Auto-Encoders, Take 1

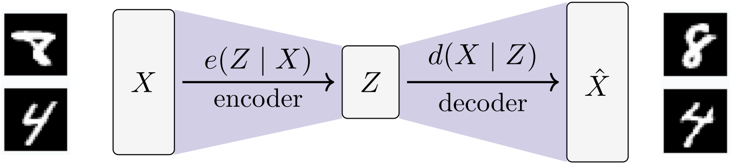
- Structure consists of two neural networks (cpds):



- Objective:

Variational Auto-Encoders, Take 1

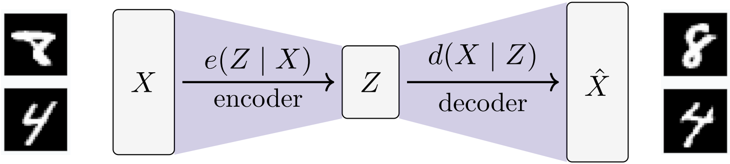
- Structure consists of two neural networks (cpds):



- Objective:
  - For each x, want to minimize $\mathrm{Rec}(x) := \overbrace{- \mathop{\mathbb{E}}_{z \sim e|x} \log d(x \mid z)}^{\text{“reconstruction error”}}$

Variational Auto-Encoders, Take 1

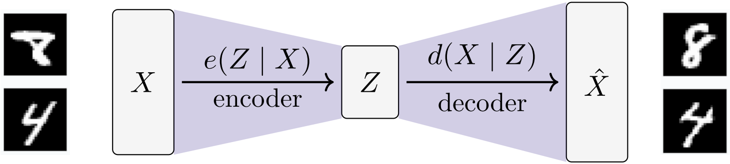
- Structure consists of two neural networks (cpds):



- Objective:
  - For each x, want to minimize $\mathrm{Rec}(x) := \overset{\text{“reconstruction error”}}{-\underset{z\sim e|x}{\mathbb{E}} \log d(x \mid z)}$
  - Also have a distribution $p(Z)$ that we want encodings to match.

Variational Auto-Encoders, Take 1

- Structure consists of two neural networks (cpds):

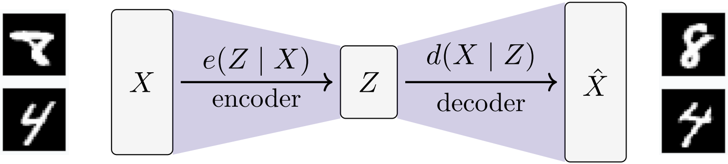


- Objective:
  - For each x, want to minimize $\mathrm{Rec}(x) := -\overset{\text{“reconstruction error”}}{\mathop{\mathbb{E}}_{z \sim e|x}} \log d(x \mid z)$
  - Also have a distribution $p(Z)$ that we want encodings to match.
  - Combine to get VaE objective:

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

Variational Auto-Encoders, Take 1

- Structure consists of two neural networks (cpds):

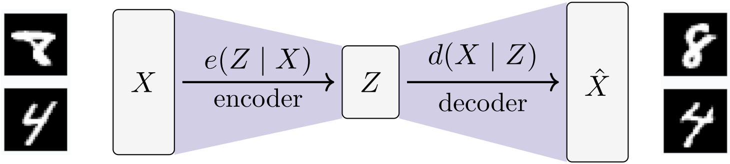


- Objective:
  - For each x, want to minimize $\mathrm{Rec}(x) := \overbrace{- \mathop{\mathbb{E}}_{z \sim e|x} \log d(x \mid z)}^{\text{"reconstruction error"}}$
  - Also have a distribution $p(Z)$ that we want encodings to match.
  - Combine to get VaE objective:

$$\mathrm{ELBO}_{p,e,d}(x) := \\ \underbrace{-\boldsymbol{D}\big(e(Z|x) \,\big\|\, p(Z)\big)}_{\text{divergence from prior}}$$

Variational Auto-Encoders, Take 1

- Structure consists of two neural networks (cpds):

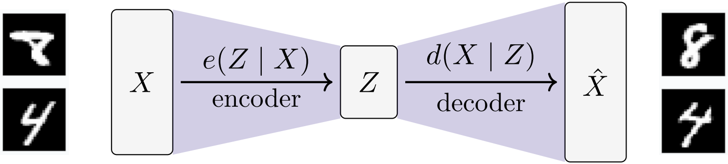


- Objective:
  - For each x, want to minimize $\mathrm{Rec}(x) := \overbrace{- \mathop{\mathbb{E}}_{z \sim e|x} \log d(x \mid z)}^{\text{“reconstruction error”}}$
  - Also have a distribution $p(Z)$ that we want encodings to match.
  - Combine to get VaE objective:

$$\mathrm{ELBO}_{p,e,d}(x) := \\ -\underbrace{\boldsymbol{D}\big(e(Z|x) \,\|\, p(Z)\big)}_{\text{divergence from prior}} - \mathrm{Rec}(x)$$

Variational Auto-Encoders, Take 1

- Structure consists of two neural networks (cpds):

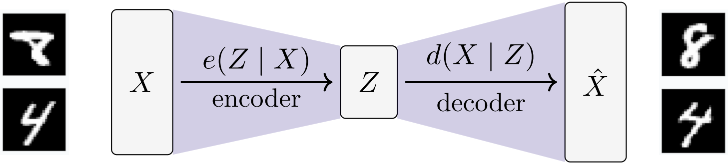


- Objective:
  - For each x, want to minimize $\mathrm{Rec}(x) := \overbrace{- \mathop{\mathbb{E}}_{z \sim e|x} \log d(x \mid z)}^{\text{“reconstruction error”}}$
  - Also have a distribution $p(Z)$ that we want encodings to match.
  - Combine to get VaE objective:

$$\mathrm{ELBO}_{p,e,d}(x) := \underbrace{-\boldsymbol{D}\big(e(Z|x) \,\|\, p(Z)\big)}_{\text{divergence from prior}} - \mathrm{Rec}(x) = \mathop{\mathbb{E}}_{z \sim e|x} \left[\log \frac{p(z) d(x \mid z)}{e(z \mid x)} \right]$$

Variational Auto-Encoders, Take 1

- Structure consists of two neural networks (cpds):

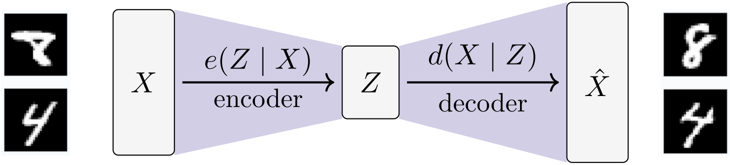


- Objective:
  - For each x, want to minimize $\mathrm{Rec}(x) := \overset{\text{"reconstruction error"}}{- \mathop{\mathbb{E}}_{z \sim e|x} \log d(x \mid z)}$
  - Also have a distribution $p(Z)$ that we want encodings to match.
  - Combine to get VaE objective:

$$\mathrm{ELBO}_{p,e,d}(x) := \underbrace{-\boldsymbol{D}\big(e(Z|x) \,\|\, p(Z)\big)}_{\text{divergence from prior}} - \mathrm{Rec}(x) = \mathop{\mathbb{E}}_{z \sim e|x} \left[\log \frac{p(z) d(x \mid z)}{e(z \mid x)} \right] \leq \overset{\text{"evidence"}}{\log \Pr_{pd}(x)}$$

Variational Auto-Encoders, Take 1

- Structure consists of two neural networks (cpds):



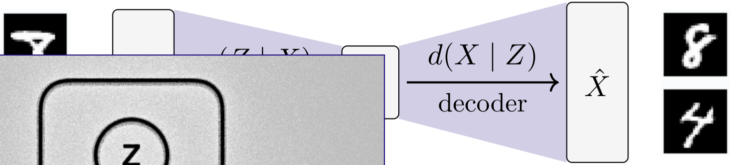
- Objective:
  - For each x, want to minimize $\mathrm{Rec}(x) := \overbrace{-\mathop{\mathbb{E}}_{z\sim e|x} \log d(x \mid z)}^{\text{“reconstruction error”}}$
  - Also have a distribution $p(Z)$ that we want encodings to match.
  - Combine to get VaE objective:

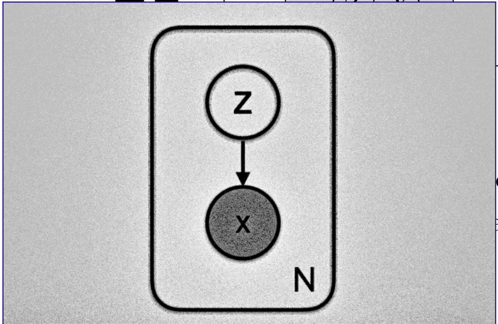
$$\mathrm{ELBO}_{p,e,d}(x) := \\ \underbrace{-\boldsymbol{D}\big(e(Z|x) \,\big\|\, p(Z)\big)}_{\text{divergence from prior}} - \mathrm{Rec}(x) = \mathop{\mathbb{E}}_{z\sim e|x}\left[\log \frac{p(z)d(x \mid z)}{e(z \mid x)}\right] \leq \overbrace{\log \Pr_{pd}(x)}^{\text{“evidence”}}$$

Urge to use graphical models (even if can't quite capture *entire* VaE)

Variational Auto-Encoders, Take 1

- Structure consists of two neural networks (cpds):





THE GRAPHICAL MODEL REPRESENTATION OF THE MODEL IN THE VARIATIONAL AUTOENCODER. THE LATENT VARIABLE Z IS A STANDARD NORMAL, AND THE DATA ARE DRAWN FROM P(X|Z). THE SHADED NODE FOR X DENOTES OBSERVED DATA. FOR BLACK AND WHITE IMAGES OF HANDWRITTEN DIGITS, THIS DATA LIKELIHOOD IS BERNOULLI DISTRIBUTED.

"reconstruction error"

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

at we want encodings to match.

"evidence"

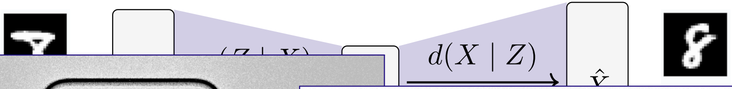
$$\mathop{\mathbb{E}}_{\sim e|x} \left[\log \frac{p(z) d(x \mid z)}{e(z \mid x)} \right] \leq \log \Pr_{pd}(x)$$

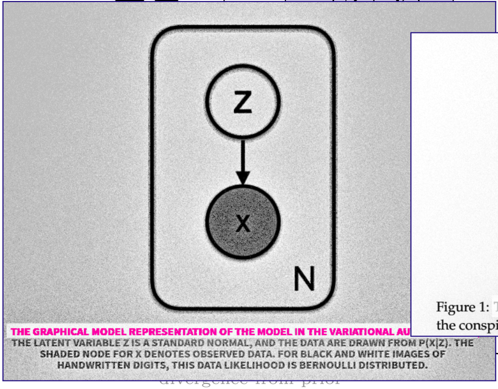
divergence from prior

Urge to use graphical models (even if can't quite capture *entire* VaE)

Variational Auto-Encoders, Take 1

- Structure consists of two neural networks (cpds):





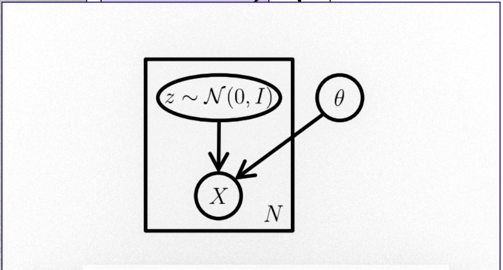
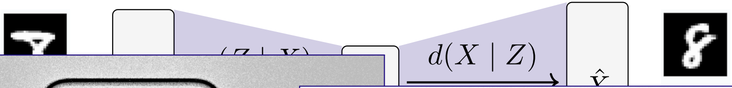


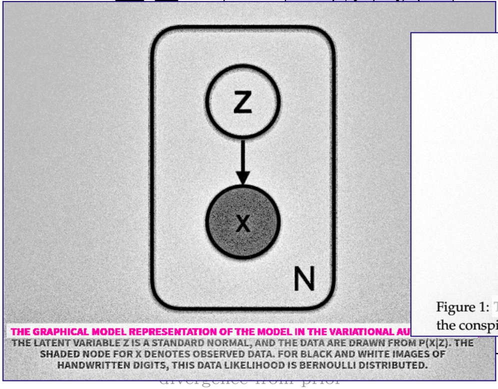
Figure 1: The standard VAE model represented as a graphical model. Note the conspicuous lack of any structure or even an "encoder" pathway: it is

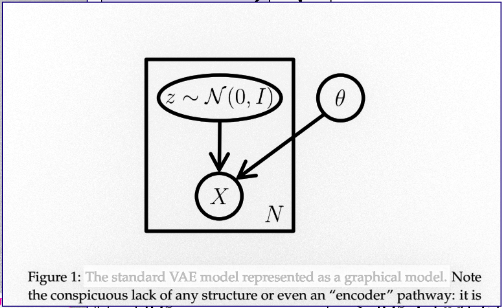
Urge to use graphical models (even if can't quite capture *entire* VaE)

Variational Auto-Encoders, Take 1

- Structure consists of two neural networks (cpds):





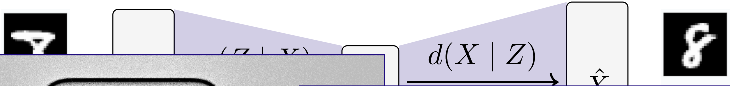


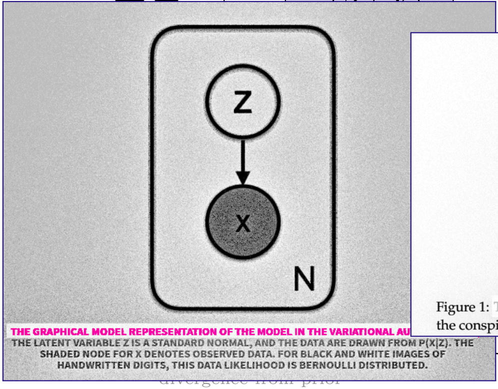
Urge to use graphical models (even if can't quite capture *entire* VaE)

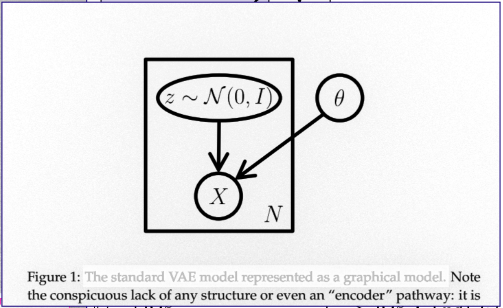
- $e(Z \mid X)$ has same target as $p(Z)$, so can't put in BN;

Variational Auto-Encoders, Take 1

- Structure consists of two neural networks (cpds):





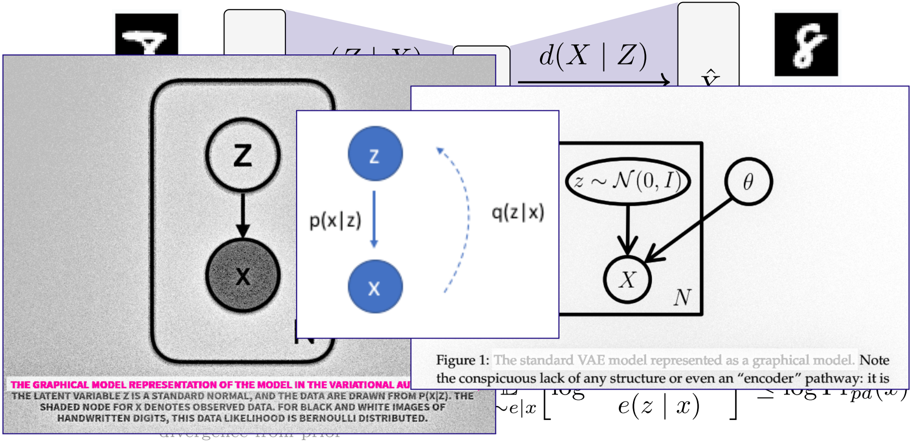


Urge to use graphical models (even if can't quite capture *entire* VaE)

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

Variational Auto-Encoders, Take 1

- Structure consists of two neural networks (cpds):



Urge to use graphical models (even if can't quite capture *entire* VaE)

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

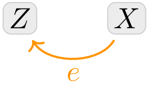
Variational Auto-Encoders, Take 2

- Structure:

$$Z \qquad X$$

Variational Auto-Encoders, Take 2

- Structure:
  $e(Z \mid X)$: encoder

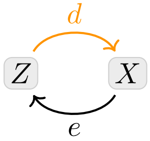


Variational Auto-Encoders, Take 2

- Structure:

  $e(Z \mid X)$: encoder

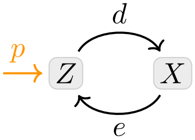
  $d(X \mid Z)$: decoder



Variational Auto-Encoders, Take 2

- Structure:

  $e(Z \mid X)$: encoder
  $d(X \mid Z)$: decoder
  $p(Z)$: prior



Variational Auto-Encoders, Take 2

- Structure:

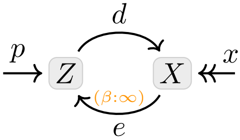
  $e(Z \mid X)$: encoder
  $d(X \mid Z)$: decoder
  $p(Z)$: prior

- observe a sample x

$$\begin{tikzcd} {} \arrow[r, "p"] & Z \arrow[r, bend left, "d"] & X \arrow[l, bend left, "e"] & {} \arrow[l, two heads, "x"'] \end{tikzcd}$$

Variational Auto-Encoders, Take 2

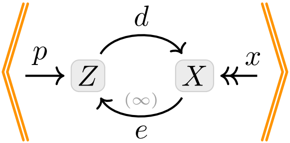
- Structure:
  $e(Z \mid X)$: encoder
  $d(X \mid Z)$: decoder
  $p(Z)$: prior
- observe a sample x



Variational Auto-Encoders, Take 2

- Structure:
  $e(Z \mid X)$: encoder
  $d(X \mid Z)$: decoder
  $p(Z)$: prior
- observe a sample x

Objective function is free:

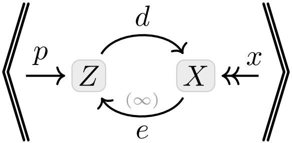


$= \mathrm{ELBO}_{p,e,d}(x)$

Variational Auto-Encoders, Take 2

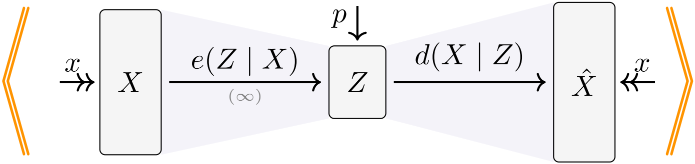
- Structure:
  - $e(Z \mid X)$: encoder
  - $d(X \mid Z)$: decoder
  - $p(Z)$: prior
- observe a sample x

Objective function is free:



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

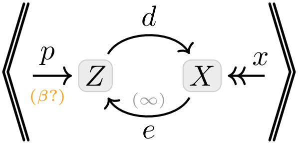
$=$



Variational Auto-Encoders, Take 2

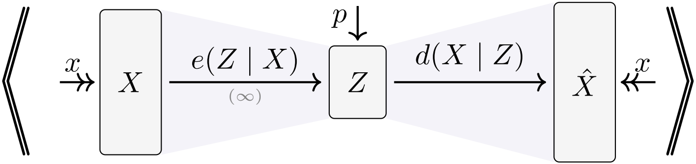
- Structure:
  $e(Z \mid X)$: encoder
  $d(X \mid Z)$: decoder
  $p(Z)$: prior
- observe a sample x

Objective function is free:



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

$=$



Visual Proof: The Variational Bound

Another Visual Proof: Data Processing Inequality

Visual Proof: The Variational Bound

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

▸ Another Visual Proof: Data Processing Inequality

Visual Proof: The Variational Bound

$$-\log \Pr_{p,d}(X{=}x) \;=\;$$

$$\left\langle\!\!\left\langle \xrightarrow{p} \boxed{Z} \xrightarrow{d} \boxed{X} \overset{x}{\twoheadleftarrow} \right\rangle\!\!\right\rangle \qquad \left\langle\!\!\left\langle \begin{tikzcd} {} \arrow[r, "p"] & \boxed{Z} \arrow[r, bend left, "d"] & \boxed{X} \arrow[l, bend left, "\infty", "e"'] & {} \arrow[l, two heads, "x"'] \end{tikzcd} \right\rangle\!\!\right\rangle$$

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

▸ Another Visual Proof: Data Processing Inequality

Visual Proof: The Variational Bound

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

Another Visual Proof: Data Processing Inequality

Recap

PDGs. . .

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

Recap

PDGs. . .

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

Recap

PDGs...

- capture inconsistency, including conflicting information from multiple sources with varying reliability.
- are especially modular.
- cleanly separate quantitative from qualitative information, and can express variable confidence in each (β and α).

Recap

PDGs...

- capture inconsistency, including conflicting information from multiple sources with varying reliability.
- are especially modular.
- cleanly separate quantitative from qualitative information, and can express variable confidence in each (β and α).
- naturally capture BNs and factor graphs, with the best-scoring distribution.

Recap

PDGs...

- capture inconsistency, including conflicting information from multiple sources with varying reliability.
- are especially modular.
- cleanly separate quantitative from qualitative information, and can express variable confidence in each (β and α).
- naturally capture BNs and factor graphs, with the best-scoring distribution.
- simultaneously capture many standard loss functions and divergences with the value of the scoring function.

Recap

PDGs...

- capture inconsistency, including conflicting information from multiple sources with varying reliability.
- are especially modular.
- cleanly separate quantitative from qualitative information, and can express variable confidence in each (β and α).
- naturally capture BNs and factor graphs, with the best-scoring distribution.
- simultaneously capture many standard loss functions and divergences with the value of the scoring function.
- give us a clean visual language for reasoning about the relationships between objectives.

Recap

PDGs...

- capture inconsistency, including conflicting information from multiple sources with varying reliability.
- are especially modular.
- cleanly separate quantitative from qualitative information, and can express variable confidence in each (β and α).
- naturally capture BNs and factor graphs, with the best-scoring distribution.
- simultaneously capture many standard loss functions and divergences with the value of the scoring function.
- give us a clean visual language for reasoning about the relationships between objectives.

But there is much more to be done!

Outline for Section 8

$$
\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "k"] \\
C \arrow[r, "h"'] & D
\end{tikzcd}
$$

More Category Theory

Commutative diagram
$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "k"] \\ C \arrow[r, "h"'] & D \end{tikzcd}$$
says $\forall a.\ kfa = hga.$

▸ More Category Theory

$$\text{PDG} \quad m := \begin{tikzcd} A \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "k"] \\ C \arrow[r, "h"'] & D \end{tikzcd}$$

PDG inconsistency measures how close the diagram is to commuting

$$\exp\left(-\frac{1}{\gamma}\langle\!\langle m \rangle\!\rangle_\gamma\right) = \#\left\{a : kfa = hga\right\}$$

‣ More Category Theory

Databases

Database

$\mathbf{R}_1$

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

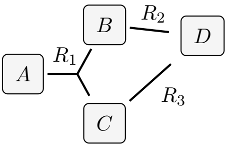
$\mathbf{R}_2$

B	D
b_2	d_1
b_3	d_2
b_4	d_3

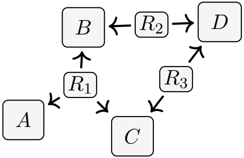
$\mathbf{R}_3$

C	D
c_2	d_1
c_1	d_3

Relational Schema



Row-PDG



Proposition

If $\mathcal{D}$ is a database and μ is a joint distribution over $\mathcal{M}_{\mathcal{D}}$, then $\mu \in [\![\mathcal{M}_{\mathcal{D}}]\!]$ iff $\mathrm{Supp}(\mu)$ is a universal relation for $\mathcal{D}$.

Corollary

$\mathcal{M}_{\mathcal{D}}$ is consistent iff $\mathcal{D}$ is join consistent.

Ongoing and Future Work

- Belief propagation as local inconsistency reduction

Ongoing and Future Work

- Belief propagation as local inconsistency reduction
- Properties of "sub-stochastic" PDGs: semantics for incomplete cpds

Ongoing and Future Work

- Belief propagation as local inconsistency reduction
- Properties of “sub-stochastic” PDGs: semantics for incomplete cpds
- Trace Semantics and Composition

Ongoing and Future Work

- Belief propagation as local inconsistency reduction
- Properties of “sub-stochastic” PDGs: semantics for incomplete cpds
- Trace Semantics and Composition
  - Extend semantics to score other objects, not just joint distributions.

Ongoing and Future Work

- Belief propagation as local inconsistency reduction
- Properties of “sub-stochastic” PDGs: semantics for incomplete cpds
- Trace Semantics and Composition
    - Extend semantics to score other objects, not just joint distributions.
    - Regarding PDGs as probabilistic automata.

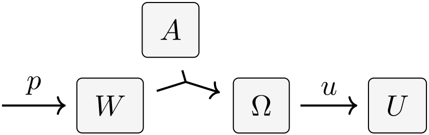
Ongoing and Future Work

- Belief propagation as local inconsistency reduction
- Properties of "sub-stochastic" PDGs: semantics for incomplete cpds
- Trace Semantics and Composition
  - Extend semantics to score other objects, not just joint distributions.
  - Regarding PDGs as probabilistic automata.
  - Open Question: Do PDGs capture Dependency Networks? $*$

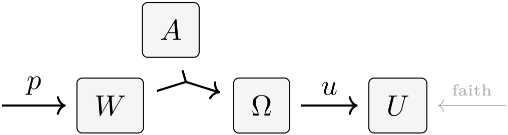
Ongoing and Future Work

- Belief propagation as local inconsistency reduction
- Properties of "sub-stochastic" PDGs: semantics for incomplete cpds
- Trace Semantics and Composition
  - Extend semantics to score other objects, not just joint distributions.
  - Regarding PDGs as probabilistic automata.
  - Open Question: Do PDGs capture Dependency Networks? $*$
- Encoding preferences, and understanding preference changes

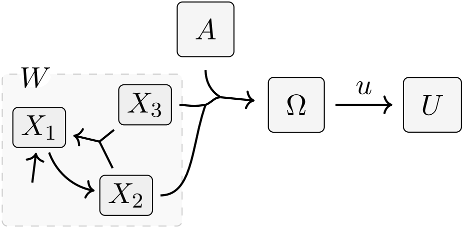
A Different Flavor of Agent



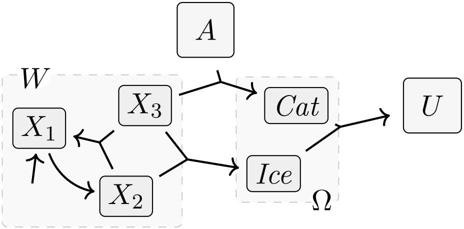
A Different Flavor of Agent



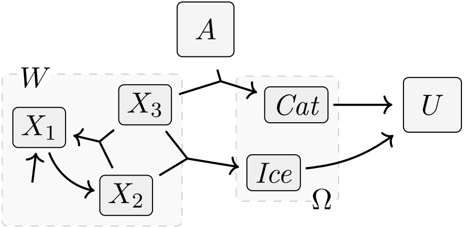
A Different Flavor of Agent



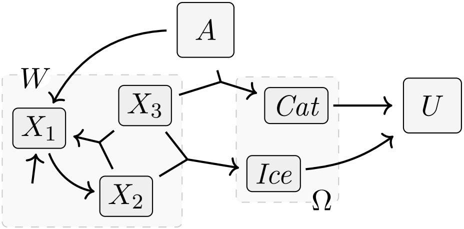
A Different Flavor of Agent



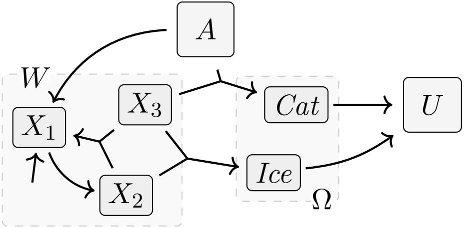
A Different Flavor of Agent



A Different Flavor of Agent

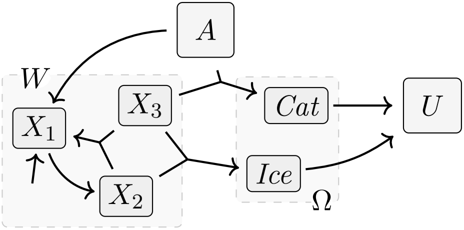


A Different Flavor of Agent



- Driven by pursuit of coherence;
  not (necessarily) maximization.

A Different Flavor of Agent

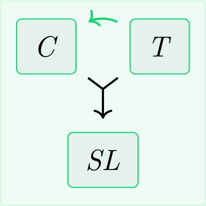


- Driven by pursuit of coherence;
  not (necessarily) maximization.

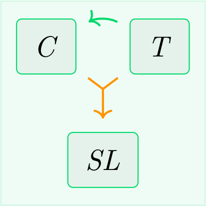
PDG Python library available at
`https://orichardson.github.io/pdg/`

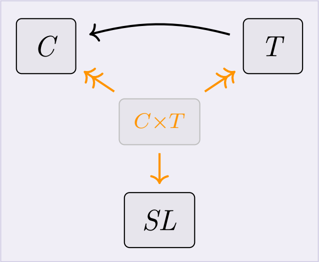
Outline for Section 9

Hyper-graphs? Or merely graphs?



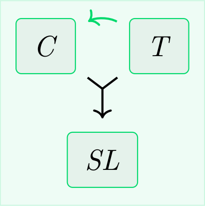
Hyper-graphs? Or merely graphs?

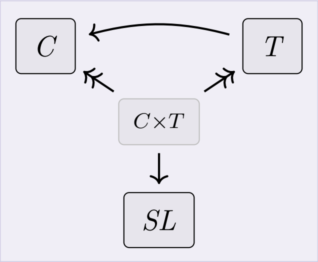




◂ main definition

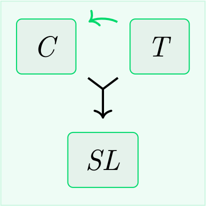
Hyper-graphs? Or merely graphs?

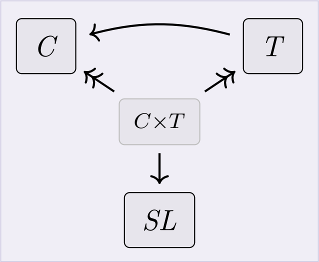




- This widget expands state space, but graphs are simpler.

Hyper-graphs? Or merely graphs?



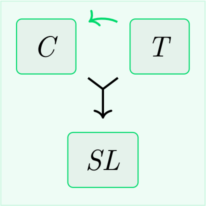


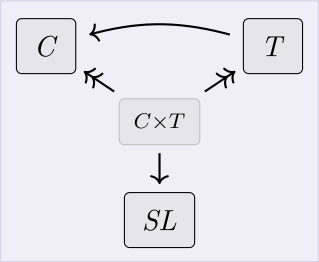
- This widget expands state space, but graphs are simpler.
- There is a natural correspondence

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

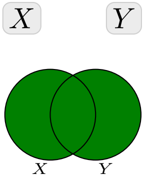
joint distributions $\leftrightarrows$ expanded joint distributions satisfying coherence constraints

(working directly with hypergraphs is also possible)

main definition

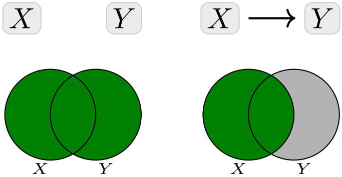
Outline for Section 10

Illustrations of *IDef*



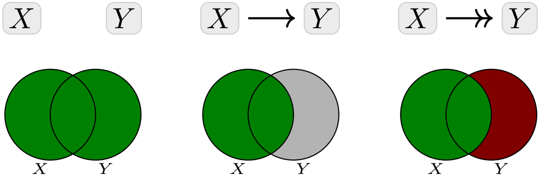
◂ back to semantics

Illustrations of *IDef*



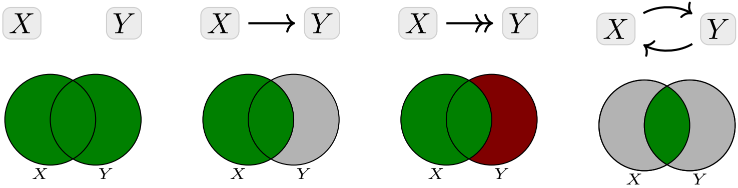
back to semantics

ILLUSTRATIONS OF *IDef*

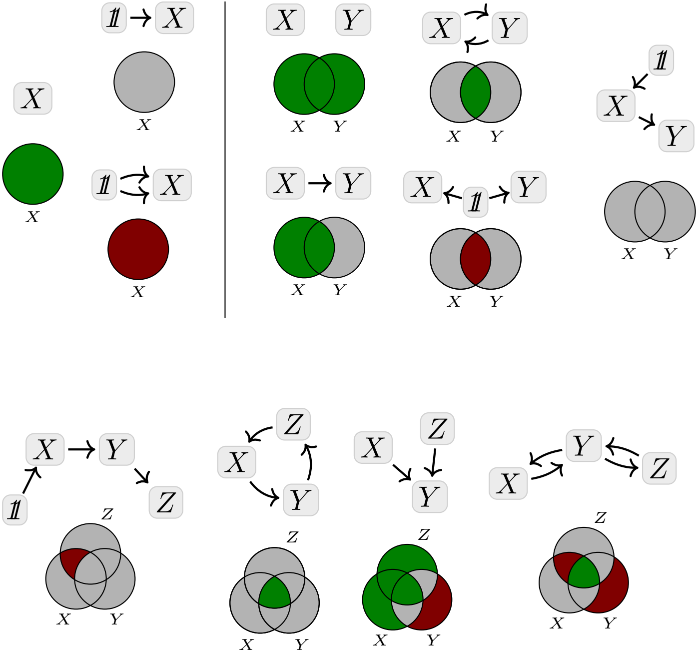


◂ back to semantics

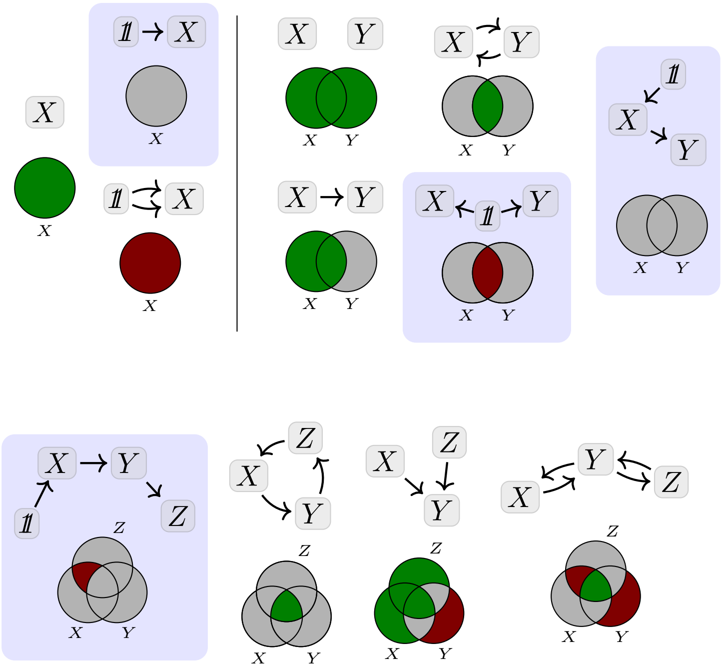
ILLUSTRATIONS OF *IDef*



◂ back to semantics



back to semantics



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

Outline for Section 11

Proposition (uniqueness for small γ)

1. *If* $0 < \gamma \leq \min_L \beta_L^{\mathcal{m}}$, *then* $[\![\mathcal{m}]\!]_\gamma^*$ *is a singleton.*
2. $\lim\limits_{\gamma \to 0} [\![\mathcal{m}]\!]_\gamma^*$ *exists and is a singleton.*

Proposition (uniqueness for small γ)

1. *If* $0 < \gamma \leq \min_L \beta_L^{m}$, *then* $[\![m]\!]_\gamma^*$ *is a singleton.*
2. $\lim\limits_{\gamma \to 0} [\![m]\!]_\gamma^*$ *exists and is a singleton.*

This lets us define $[\![m]\!]^* :=$ unique element $\left(\lim\limits_{\gamma \to 0} [\![m]\!]_\gamma^* \right)$.

Relationships Between Semantics

Proposition (*the set of consistent distributions is the zero set of the scoring function*)

$\{\!\{ m \}\!\} = \{\mu : [\![m]\!]_0(\mu) = 0\}$.

Proposition (*If there there are distributions consistent with m, the best distribution is one of them.*)

$[\![m]\!]^* \in [\![m]\!]_0^*$, *so if* m *is consistent, then* $[\![m]\!]^* \in \{\!\{ m \}\!\}$.

back to semantics properties

Another View of PDG semantics

$$[\![m]\!]_\gamma(\mu) = \mathop{\mathbb{E}}_{\mu} \log \prod_{X \xrightarrow{L} Y} \left(\frac{\mu(Y \mid X)}{\mathbf{p}_L(Y \mid X)} \right)^{\beta_L} \left(\frac{\mu(\mathcal{N})}{\prod_{X \xrightarrow{L} Y} \mu(Y \mid X)^{\alpha_L}} \right)^{\gamma}$$

Comparing PDG to Factor Graph Semantics

$$[\![m]\!](\mu) = \mathbb{E}_\mu \sum_{X \xrightarrow{L} Y} \Bigg[\overbrace{\beta_L \log \frac{1}{\mathbf{p}_L(Y|X)}}^{\text{log likelihood / cross entropy}} + \underbrace{(\alpha_L \gamma - \beta_L) \log \frac{1}{\mu(Y|X)}}_{\text{local regularization (if } \beta_L > \alpha_L\gamma)} \Bigg] - \overbrace{\gamma\, \mathrm{H}(\mu)}^{\text{global regularization}}.$$

Comparing PDG to Factor Graph Semantics

$$[\![m]\!](\mu) = \mathbb{E}_\mu \sum_{X \xrightarrow{L} Y} \Bigg[\overbrace{\beta_L \log \frac{1}{\mathbf{p}_L(Y|X)}}^{\text{log likelihood / cross entropy}} + \underbrace{(\alpha_L \gamma - \beta_L) \log \frac{1}{\mu(Y|X)}}_{\text{local regularization (if } \beta_L > \alpha_L \gamma)} \Bigg] - \overbrace{\gamma\, \mathrm{H}(\mu)}^{\text{global regularization}}.$$

And the weighted factor graph's canonical scoring function:

$$\mathit{VFE}_\Psi(\mu) := \mathop{\mathbb{E}}_{\mu} \left[\sum_{J \in \mathcal{J}} \theta_J \log \frac{1}{\phi_J(X_J)} \right] - \mathrm{H}(\mu)$$

Properties of Inconsistency, for minimization

$$\langle\!\langle \mathcal{m} \rangle\!\rangle_\gamma := \inf_\mu [\![\mathcal{m}]\!]_\gamma$$

Nice properties for minimization:

- The function $\gamma \mapsto \langle\!\langle \mathcal{m} \rangle\!\rangle_\gamma$ is continuous for all γ
- The function $p \mapsto \langle\!\langle \mathcal{m} \sqcup p \rangle\!\rangle_\gamma$ is smooth and strictly convex on its interior.

$$VFE_\Phi(\mu) := \mathop{\mathbb{E}}_{\mu}\Big[-\sum_{J\in\mathcal{J}} \theta_J \log \phi_J(X_J) \Big] - \mathrm{H}(\mu)$$

◂ Back to Factor Graph Definition

Outline for Section 12

Bayesian Networks: Maximum Entropy?

Common distributions tend to maximize entropy subject to natural constraints.

distribution	constraints
Gaussian $\mathcal{N}(\mu, \sigma^2)$	mean μ, variance σ^2
Exponential $\mathrm{Exp}(\lambda)$	positive support, mean λ
Factor graphs	moment matching.
$\dots$	$\dots$

back to BN theorem

Bayesian Networks: Maximum Entropy?

Common distributions tend to maximize entropy subject to natural constraints.

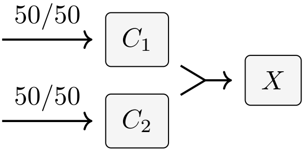
distribution	constraints
Gaussian $\mathcal{N}(\mu, \sigma^2)$	mean μ, variance σ^2
Exponential $\mathrm{Exp}(\lambda)$	positive support, mean λ
Factor graphs	moment matching.
$\ldots$	$\ldots$
Bayesian Networks	cpds + ???

back to BN theorem

Bayesian Networks: Maximum Entropy?

Common distributions tend to maximize entropy subject to natural constraints.

distribution	constraints
Gaussian $\mathcal{N}(\mu, \sigma^2)$	mean μ, variance σ^2
Exponential $\text{Exp}(\lambda)$	positive support, mean λ
Factor graphs	moment matching.
...	...
Bayesian Networks	cpds + ???



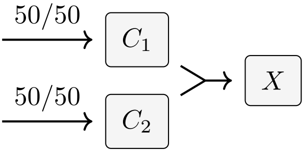
$$\Pr(X \mid C_1, C_2) = \begin{cases} \mathbb{1}[X = x_0] & \text{if } C_1 = C_2 \\ \text{Unif}(X) & \text{if } C_1 \neq C_2 \end{cases}$$

◂ back to BN theorem

Bayesian Networks: Maximum Entropy?

Common distributions tend to maximize entropy subject to natural constraints.

distribution	constraints
Gaussian $\mathcal{N}(\mu, \sigma^2)$	mean μ, variance σ^2
Exponential $\mathrm{Exp}(\lambda)$	positive support, mean λ
Factor graphs	moment matching.
$\ldots$	$\ldots$
Bayesian Networks	cpds + ???



$$\Pr(X \mid C_1, C_2) = \begin{cases} \mathbb{1}[X = x_0] & \text{if } C_1 = C_2 \\ \mathrm{Unif}(X) & \text{if } C_1 \neq C_2 \end{cases}$$

Corollary

Among the distributions in $\{\!\{\mathcal{B}\}\!\}$, $\Pr_{\mathcal{B}}$ *has the maximum entropy, beyond the entropy of the given cpds.*

IDef says maximize: $$\mathrm{H}(\mu) - \sum_{X \in \mathcal{N}} \mathrm{H}_\mu(X \mid \mathbf{Pa}\, X)$$

back to BN theorem

Full Factor Graph Results

Theorem (PDGs are WFGs)

If $\boldsymbol{\beta} = \gamma\boldsymbol{\alpha}$*, then* $[\![\mathcal{M}]\!]^*_\gamma = \Pr_{(\Phi_{\mathcal{M}}, \boldsymbol{\beta})}$.
Concretely, for all unweighted PDGs $\mathcal{N}$ *and non-negative vectors* $\mathbf{v}$ *over the edges of* $\mathcal{N}$*, and all* $\gamma > 0$*, we have that* $[\![(\mathcal{N}, \mathbf{v}, \gamma\mathbf{v})]\!]_\gamma = \gamma\, \mathit{VFE}_{(\Phi_{\mathcal{N}}, \mathbf{v})}$*; consequently,* $[\![(\mathcal{N}, \mathbf{v}, \gamma\mathbf{v})]\!]^*_\gamma = \{\Pr_{(\Phi_{\mathcal{N}}, \mathbf{v})}\}$.

Theorem (WFGs are PDGs)

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

◂ PDG→WFG ◂ WFG→PDG

Outline for Section 13

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

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

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

$$\mathrm{I}_p(x) = \log \frac{1}{p(x)} = \left\langle\!\!\left\langle \begin{tikzcd} & \cdot \arrow[dl] \arrow[dr, "p"] & & \\ Z & & X & \arrow[l, two heads, "x"'] \end{tikzcd} \right\rangle\!\!\right\rangle.$$

Proposition (supervised setting: conditional cross entropy)

The inconsistency of the PDG containing $f(Y \mid X)$ and a high-confidence empirical distribution $\mathrm{Pr}_{\underline{\mathbf{xy}}}$ of samples $\underline{\mathbf{xy}} = \{(x_i, y_i)\}$ is equal to the cross entropy (plus $\mathrm{H}(Y \mid X)$, a constant that depends only on the data $\mathrm{Pr}_{\underline{\mathbf{xy}}}$). That is,

$$\left\langle\!\!\left\langle \begin{tikzcd} & \mathrm{Pr}_{\underline{\mathbf{xy}}} \arrow[dl] \arrow[dr, "(\beta:\infty)"] & \\ X \arrow[rr, "f"'] & & Y \end{tikzcd} \right\rangle\!\!\right\rangle = \frac{1}{|\underline{\mathbf{xy}}|} \sum_{(x,y)\in\underline{\mathbf{xy}}} \left[\log \frac{1}{f(y \mid x)} \right] \quad - \mathrm{H}_{\mathrm{Pr}_{\underline{\mathbf{xy}}}}(Y \mid X).$$

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

$$\left\langle\!\!\left\langle \xrightarrow[(\beta)]{D} \boxed{X} \begin{array}{c} \overset{h}{\twoheadrightarrow} \\ \underset{f}{\twoheadrightarrow} \end{array} \boxed{Y} \right\rangle\!\!\right\rangle = -\beta \log\Big(\mathrm{accuracy}_{f,D}(h)\Big) = \beta\ \mathrm{I}_D[f = h].$$

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

$$\left\langle\!\!\left\langle \xrightarrow[(\beta)]{D} \boxed{X} \underset{f}{\overset{h}{\rightrightarrows}} \boxed{Y} \right\rangle\!\!\right\rangle = -\beta \log\Big(\text{accuracy}_{f,D}(h)\Big) = \beta\ \mathrm{I}_D[f = h].$$

- Thought of as a feature of h, but as a PDG, symmetry between f, h is clear.

Proposition (Mean Square Error as Inconsistency)

$$\left\langle\!\!\left\langle \begin{tikzcd} {} \arrow[r, "D", "(\beta:\infty)"'] & X \arrow[r, bend left, "\mathcal{N}(f(x),1)"] \arrow[r, bend right, "\mathcal{N}(g(x),1)"'] & Y \end{tikzcd} \right\rangle\!\!\right\rangle = \mathbb{E}_D \Big(f(X) - h(X)\Big)^2 =: \mathrm{MSE}(f,h)$$

Proposition (Regularizers as priors)

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

That is,

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

◂ back to table of losses

Proposition (Regularizers as priors)

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

That is,

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

◂ back to table of losses

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,

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

◂ back to table of losses

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,

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

back to table of losses

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,

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

back to table of losses

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,

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

back to table of losses

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,

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

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

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

back to table of losses

Surprise as Inconsistency

Consider a distribution $p(X)$.

The surprise (information content) at seeing a sample x is:

$$\mathrm{I}_p(x) := \log \frac{1}{p(X\!=\!x)}.$$

Proposition

Average Surprise is the inconsistency of simultaneously believing p and an emperical distribution $\Pr_{\underline{\mathbf{x}}}$, with high confidence (plus $\mathrm{H}(Y \mid X)$, a constant that depends only on the data $\Pr_{\underline{\mathbf{xy}}}$) That is,

$$\mathrm{I}_p(x) = \left\langle\!\!\left\langle \xrightarrow{p} \boxed{X} \twoheadleftarrow\!\!\xrightarrow[(\beta:\infty)]{\Pr_{\underline{\mathbf{x}}}} \right\rangle\!\!\right\rangle + \mathrm{H}(\Pr_{\underline{\mathbf{x}}}).$$

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

For entire dataset

Outline for Section 14

Visual Proof: Data Processing Inequality

$$\boldsymbol{D}^{\mathrm{PDG}}_{(\beta,\zeta)}\Big(p \,\Big\|\, q\Big) \geq \boldsymbol{D}^{\mathrm{PDG}}_{(\beta,\zeta)}\Big(f \circ p \,\Big\|\, f \circ q\Big)$$

$$\left\langle\!\!\left\langle \xrightarrow[(\beta)]{p} \boxed{X} \xleftarrow[(\zeta)]{q} \right\rangle\!\!\right\rangle$$

$$\left\langle\!\!\left\langle \xrightarrow[(\beta)]{f\circ p} \boxed{X} \xleftarrow[(\zeta)]{f\circ q} \right\rangle\!\!\right\rangle$$

back to variational bound proof

Visual Proof: Data Processing Inequality

$$\boldsymbol{D}^{\mathrm{PDG}}_{(\beta,\zeta)}\Big(p \,\Big\|\, q\Big) \geq \boldsymbol{D}^{\mathrm{PDG}}_{(\beta,\zeta)}\Big(f \circ p \,\Big\|\, f \circ q\Big)$$

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

$$\left\langle\!\!\left\langle \xrightarrow[(\beta)]{f\circ p} \boxed{X} \xleftarrow[(\zeta)]{f\circ q} \right\rangle\!\!\right\rangle$$

◂ back to variational bound proof

Visual Proof: Data Processing Inequality

$$\boldsymbol{D}^{\mathrm{PDG}}_{(\beta,\zeta)}\left(p \,\middle\|\, q\right) \geq \boldsymbol{D}^{\mathrm{PDG}}_{(\beta,\zeta)}\left(f\circ p \,\middle\|\, f\circ q\right)$$

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

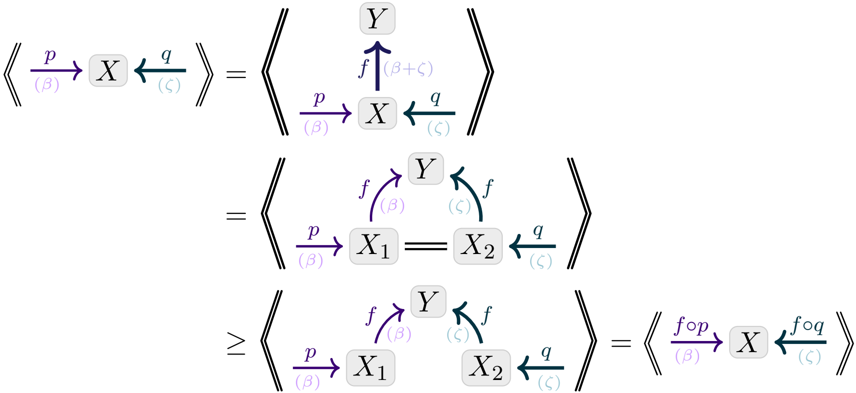
$$= \left\langle\!\!\left\langle \begin{tikzcd} & & \boxed{Y} & & \\ {} \arrow[r, "p", "(\beta)"'] & \boxed{X_1} \arrow[ur, bend left, "f", "(\beta)"'] \arrow[rr, equal] & & \boxed{X_2} \arrow[ul, bend right, "f"', "(\zeta)"] & {} \arrow[l, "q"', "(\zeta)"] \end{tikzcd} \right\rangle\!\!\right\rangle$$

$$\left\langle\!\!\left\langle \xrightarrow[(\beta)]{f\circ p} \boxed{X} \xleftarrow[(\zeta)]{f\circ q} \right\rangle\!\!\right\rangle$$

back to variational bound proof

Visual Proof: Data Processing Inequality

$$\boldsymbol{D}^{\mathrm{PDG}}_{(\beta,\zeta)}\left(p \,\middle\|\, q\right) \geq \boldsymbol{D}^{\mathrm{PDG}}_{(\beta,\zeta)}\left(f \circ p \,\middle\|\, f \circ q\right)$$



◂ back to variational bound proof

Outline for Section 15

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

Back to Commutative Diagrams

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

Back to Commutative Diagrams

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.

Back to Commutative Diagrams

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 $\alpha_L = \beta_L = 1$.

Back to Commutative Diagrams

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)

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

Back to Commutative Diagrams

Definition (PDG)

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

$\mathcal{V} : \mathcal{N} \to \textbf{Set}$ (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)

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

Equivalent Categorical Definition

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

Back to Commutative Diagrams

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.

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

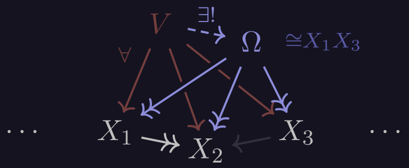
What do you do with diagrams? Take limits / colimits.

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

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

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

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



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

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

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

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

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

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

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

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

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

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

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

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

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