

Compositional inference for Bayesian networks and causality

Bart Jacobs, **Márk Széles**, Dario Stein

Computing with Markov Categories Workshop
Tallinn, February 26, 2025

Radboud Universiteit



Outline

- Discrete probability subdistributions
- Copy-discard-compare categories
- Normalisation structures in copy-discard categories
- Disintegration and Bayesian inversion
- A compositional calculus of conditioning

Discrete probability subdistributions

Definition

A **discrete probability subdistribution** on a set X is a function $\omega : X \rightarrow [0, 1]$ such that

- 1 $\text{supp}(\omega) = \{x \in X : \omega(x) \neq 0\}$ is a finite set.
- 2 $\|\omega\| = \sum_{x \in X} \omega(x) \leq 1$.

Discrete probability subdistributions

Definition

A **discrete probability subdistribution** on a set X is a function $\omega : X \rightarrow [0, 1]$ such that

- 1 $\text{supp}(\omega) = \{x \in X : \omega(x) \neq 0\}$ is a finite set.
- 2 $\|\omega\| = \sum_{x \in X} \omega(x) \leq 1$.

- Write subdistributions as formal sums $\omega = \sum_{x \in X} \omega(x)|x\rangle$

Discrete probability subdistributions

Definition

A **discrete probability subdistribution** on a set X is a function $\omega : X \rightarrow [0, 1]$ such that

- 1 $\text{supp}(\omega) = \{x \in X : \omega(x) \neq 0\}$ is a finite set.
- 2 $\|\omega\| = \sum_{x \in X} \omega(x) \leq 1$.

- Write subdistributions as formal sums $\omega = \sum_{x \in X} \omega(x)|x\rangle$
- There is a monad $\mathcal{D}_{\leq} : \mathbf{Sets} \rightarrow \mathbf{Sets}$.

Discrete probability subdistributions

Definition

A **discrete probability subdistribution** on a set X is a function $\omega : X \rightarrow [0, 1]$ such that

- 1 $\text{supp}(\omega) = \{x \in X : \omega(x) \neq 0\}$ is a finite set.
- 2 $\|\omega\| = \sum_{x \in X} \omega(x) \leq 1$.

- Write subdistributions as formal sums $\omega = \sum_{x \in X} \omega(x)|x\rangle$
- There is a monad $\mathcal{D}_{\leq} : \mathbf{Sets} \rightarrow \mathbf{Sets}$.
- Write $f : X \multimap Y$ for a Kleisli-map $f : X \rightarrow \mathcal{D}_{\leq}(Y)$.

Discrete probability subdistributions

Definition

A **discrete probability subdistribution** on a set X is a function $\omega : X \rightarrow [0, 1]$ such that

- 1 $\text{supp}(\omega) = \{x \in X : \omega(x) \neq 0\}$ is a finite set.
- 2 $\|\omega\| = \sum_{x \in X} \omega(x) \leq 1$.

- Write subdistributions as formal sums $\omega = \sum_{x \in X} \omega(x)|x\rangle$
- There is a monad $\mathcal{D}_{\leq} : \mathbf{Sets} \rightarrow \mathbf{Sets}$.
- Write $f : X \multimap Y$ for a Kleisli-map $f : X \rightarrow \mathcal{D}_{\leq}(Y)$.
- Write $f(y|x)$ for $f(x)(y) \in [0, 1]$.

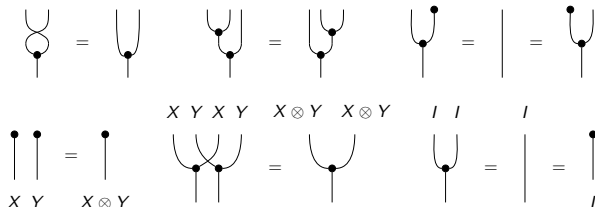
Copy-discard-compare (CDC) categories

Definition

A **copy-discard-compare category** is a symmetric monoidal category (C, \otimes, I) with **copier** $\Delta_X : X \rightarrow X \otimes X$, **discard** $d_X : X \rightarrow I$, and **comparator** $\nabla_X : X \otimes X \rightarrow X$ maps such that...

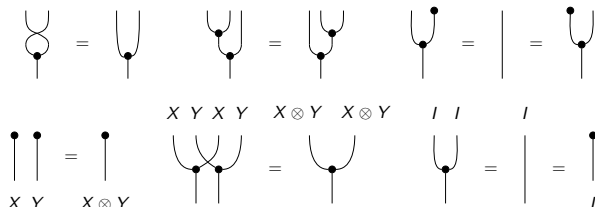
The axioms of copy-discard-compare (CDC) categories

- Copy and discard:

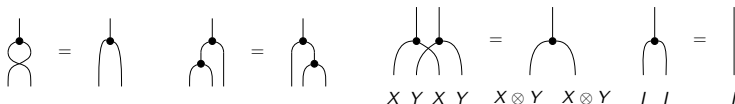


The axioms of copy-discard-compare (CDC) categories

- Copy and discard:

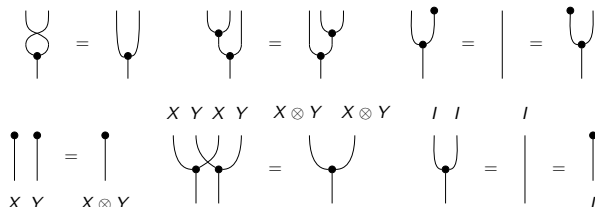


- Compare:

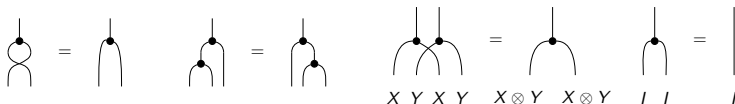


The axioms of copy-discard-compare (CDC) categories

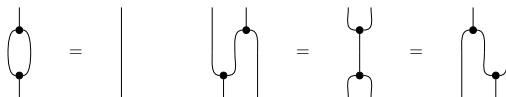
- Copy and discard:



- Compare:



- Copy-compare interaction:



Alternative definitions of CDC-categories

- CDC-categories can also be defined in terms of caps $\cap_X : X \otimes X \rightarrow I$.

A diagrammatic equation showing the multiplication of two caps. On the left, a cap with two loops (representing the product of two caps) is equal to a single cap. On the right, a cap with two loops, each labeled with X and Y , is equal to a single cap labeled with $X \otimes Y$.

Three diagrammatic equations. The first shows a cap with two loops (representing the unit) is equal to a single vertical line. The second shows a cap with a loop (representing the associativity) is equal to a single vertical line. The third shows a cap with a loop (representing the associativity) is equal to a single vertical line.

Alternative definitions of CDC-categories

- CDC-categories can also be defined in terms of caps $\cap_X : X \otimes X \rightarrow I$.



Diagrammatic equations for CDC-categories:

- A cap (a semi-circle with two downward-pointing legs) is equal to a cup (a semi-circle with two upward-pointing legs).
- A cap with two inputs labeled X and Y is equal to a cap with two inputs labeled $X \otimes Y$ and $X \otimes Y$.



Diagrammatic equations for CDC-categories:

- A cap with two inputs labeled I is equal to a vertical line with a dot at the top.
- A cap with two inputs labeled I is equal to a vertical line with a dot at the bottom.
- A cap with two inputs labeled I is equal to a cap with two inputs labeled I .

- Caps and comparators are inter-definable.



Diagrammatic equations for CDC-categories:

- A cap with two inputs labeled I is equal to a cap with two inputs labeled I .
- A cap with two inputs labeled I is equal to a cap with two inputs labeled I .

Alternative definitions of CDC-categories

- CDC-categories can also be defined in terms of caps $\cap_X : X \otimes X \rightarrow I$.

Diagrammatic equations for CDC-categories:

- A cap (a loop with two downward-pointing legs) is equal to a simple cap (a single upward arc).
- A cap with two crossings (two arcs crossing each other) is equal to a cap with two crossings (two arcs crossing each other).

Diagrammatic equations for CDC-categories:

- A cap with two crossings (two arcs crossing each other) is equal to a cap with two crossings (two arcs crossing each other).
- A cap with two crossings (two arcs crossing each other) is equal to a cap with two crossings (two arcs crossing each other).
- A cap with two crossings (two arcs crossing each other) is equal to a cap with two crossings (two arcs crossing each other).

- Caps and comparators are inter-definable.

Diagrammatic equations for CDC-categories:

- A cap (a loop with two downward-pointing legs) is equal to a cap with two crossings (two arcs crossing each other).
- A cap with two crossings (two arcs crossing each other) is equal to a cap with two crossings (two arcs crossing each other).

- There is a third, equivalent, definition via ‘least’ disintegrations of copier and identity maps.

Normalisation

- Non-zero subdistributions $\omega \in \mathcal{D}_{\leq}(X)$ can be normalised:

$$\text{nrm}(\omega) = \sum_{x \in X} \frac{\omega(x)}{\|\omega\|} |x\rangle$$

Normalisation

- Non-zero subdistributions $\omega \in \mathcal{D}_{\leq}(X)$ can be normalised:

$$\text{norm}(\omega) = \sum_{x \in X} \frac{\omega(x)}{\|\omega\|} |x\rangle$$

- Subprobability kernels $f : X \multimap Y$ can be normalised pointwise:

$$\text{norm}(f)(x) = \begin{cases} \text{norm}(f(x)) & \text{if } \|f(x)\| \neq 0 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

Normalisation

- Non-zero subdistributions $\omega \in \mathcal{D}_{\leq}(X)$ can be normalised:

$$\text{norm}(\omega) = \sum_{x \in X} \frac{\omega(x)}{\|\omega\|} |x\rangle$$

- Subprobability kernels $f : X \multimap Y$ can be normalised pointwise:

$$\text{norm}(f)(x) = \begin{cases} \text{norm}(f(x)) & \text{if } \|f(x)\| \neq 0 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

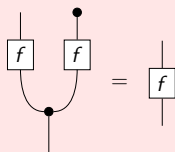
- Graphically, we write

$$\text{norm}(f) = \begin{array}{c} | \\ \boxed{f} \\ | \end{array}$$

Self-normalising maps

Definition

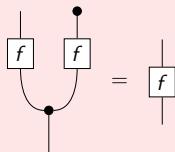
We call a map $f : X \rightarrow Y$ **self-normalising** if



Self-normalising maps

Definition

We call a map $f : X \rightarrow Y$ **self-normalising** if



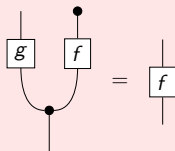
For a subchannel $f : X \multimap Y$ in $\mathcal{KI}(\mathcal{D}_{\leq})$ this translates to

$$\forall x \in X. \|f(x)\| \in \{0, 1\}$$

The 'normalised by' relation

Definition

A map $g : X \rightarrow Y$ **normalises** $f : X \rightarrow Y$ if



In this case, we write $f \preceq g$.

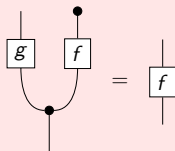
For subchannels $f, g : X \multimap Y$ in $\mathcal{KI}(\mathcal{D}_{\leq})$ this translates to

$$\forall x \in X. f(x) \neq \mathbf{0} \implies g(x) = \text{norm}(f(x))$$

The 'normalised by' relation

Definition

A map $g : X \rightarrow Y$ **normalises** $f : X \rightarrow Y$ if



In this case, we write $f \preceq g$.

For subchannels $f, g : X \multimap Y$ in $\mathcal{KI}(\mathcal{D}_{\leq})$ this translates to

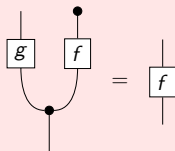
$$\forall x \in X. f(x) \neq \mathbf{0} \implies g(x) = \text{norm}(f(x))$$

- The relation \preceq is a partial order on self-normalising maps.

The 'normalised by' relation

Definition

A map $g : X \rightarrow Y$ **normalises** $f : X \rightarrow Y$ if



In this case, we write $f \preceq g$.

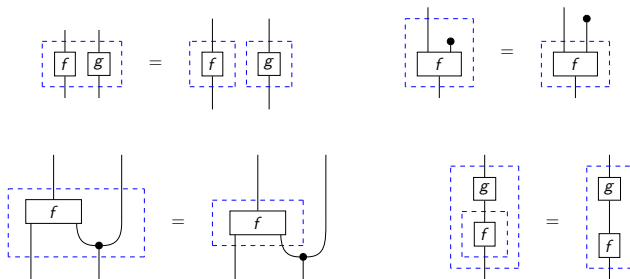
For subchannels $f, g : X \multimap Y$ in $\mathcal{KI}(\mathcal{D}_{\leq})$ this translates to

$$\forall x \in X. f(x) \neq \mathbf{0} \implies g(x) = \text{nrm}(f(x))$$

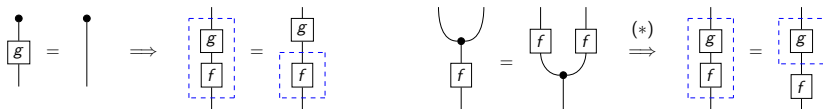
- The relation \preceq is a partial order on self-normalising maps.
- The dashed box should select the least normalisation.

Properties of normalisation

- Equational properties:



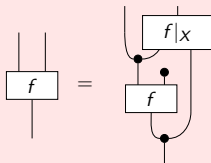
- Implicational properties:



Disintegration

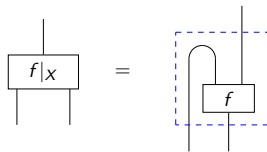
Definition

If $f : Z \rightarrow X \otimes Y$, then a **disintegration** of f is a map $f|_X : X \otimes Z \rightarrow Y$ that satisfies



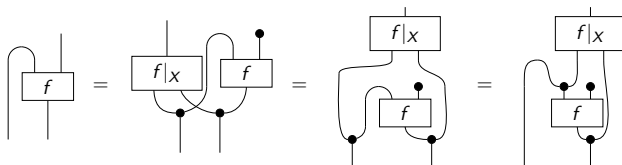
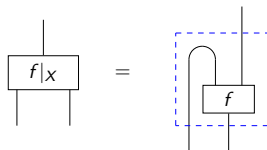
Building your own disintegrations

Try to define a disintegration by



Building your own disintegrations

Try to define a disintegration by



Building your own disintegrations

Definition

A CDC-category has **cancellative caps** if the following implication holds for all $f, g : X \rightarrow Y \otimes Z$:



The diagram shows an equality between two expressions. The left expression is a box labeled f with two input wires on the left and two output wires on the right. A curved line (a cap) connects the two input wires, forming a loop on the left. This is followed by an equals sign. The right expression is a box labeled g with two input wires on the left and two output wires on the right. A curved line (a cap) connects the two input wires, forming a loop on the left. This is followed by an implication arrow \Rightarrow . The next expression is a box labeled f with two input wires on the left and two output wires on the right, with no caps. This is followed by an equals sign. The final expression is a box labeled g with two input wires on the left and two output wires on the right, with no caps.

Building your own disintegrations

Definition

A CDC-category has **cancellative caps** if the following implication holds for all $f, g : X \rightarrow Y \otimes Z$:

$$\text{Cap}(f) = \text{Cap}(g) \Rightarrow f = g$$

Proposition (Lorenz and Tull, 2023)

In a CDC-category with normalisation and cancellative caps, every map has a disintegration.

Building your own disintegrations

Definition

A CDC-category has **cancellative caps** if the following implication holds for all $f, g : X \rightarrow Y \otimes Z$:

$$\text{Cap}(f) = \text{Cap}(g) \Rightarrow f = g$$

Proposition (Lorenz and Tull, 2023)

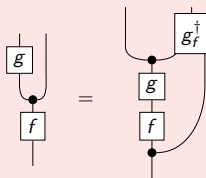
In a CDC-category with normalisation and cancellative caps, every map has a disintegration.

This does not recover disintegration in $\mathbf{BorelStoch}_{\leq 1}$!

Bayesian inversions

Definition

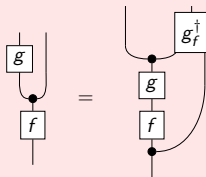
Let $g : Y \rightarrow Z$, $f : X \rightarrow Y$. A **Bayesian inversion** of g with respect to f is a map $g_f^\dagger : Z \otimes X \rightarrow Y$ that satisfies



Bayesian inversions

Definition

Let $g : Y \rightarrow Z$, $f : X \rightarrow Y$. A **Bayesian inversion** of g with respect to f is a map $g_f^\dagger : Z \otimes X \rightarrow Y$ that satisfies

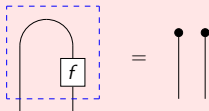


Bayesian inversions exist \iff Disintegrations exist

Broad support

Definition

Let \mathbb{C} be a CDC-category with cancellative caps. A map $f : X \rightarrow Y$ in \mathbb{C} has **broad support** if

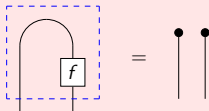


The diagram shows an equation between two expressions. On the left, a vertical line enters a square box labeled f from the bottom. A wire goes from the top of the box, loops around to the left, and then goes back down to the top of the box, forming a cap. This entire structure is enclosed in a dashed blue rectangle. To the right of this is an equals sign, followed by two parallel vertical lines, each with a solid black dot at its top end.

Broad support

Definition

Let \mathbb{C} be a CDC-category with cancellative caps. A map $f : X \rightarrow Y$ in \mathbb{C} has **broad support** if



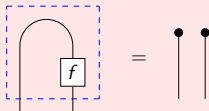
The diagram shows an equation in a string diagram calculus. On the left, a vertical line enters a square box labeled f from the bottom. A cap is formed by the line entering the box and another line that loops back to join the first line at the top. This entire structure is enclosed in a dashed blue rectangle. This is followed by an equals sign, and then two separate vertical lines, each with a dot at its top end.

- For $f : X \rightarrowtail Y$ this means $f(x)(y) > 0$ for all $x \in X, y \in Y$.

Broad support

Definition

Let \mathbb{C} be a CDC-category with cancellative caps. A map $f : X \rightarrow Y$ in \mathbb{C} has **broad support** if

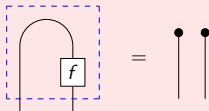


- For $f : X \rightarrowtail Y$ this means $f(x)(y) > 0$ for all $x \in X, y \in Y$.
- Maps with broad support are closed under many operations: composition, tensor, marginalisation, normalisation, disintegration.

Broad support

Definition

Let \mathbb{C} be a CDC-category with cancellative caps. A map $f : X \rightarrow Y$ in \mathbb{C} has **broad support** if

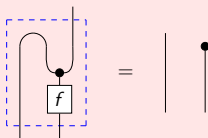


- For $f : X \rightarrowtail Y$ this means $f(x)(y) > 0$ for all $x \in X, y \in Y$.
- Maps with broad support are closed under many operations: composition, tensor, marginalisation, normalisation, disintegration.
- If a self-normalising f has broad support, then f is total.

Disappearance of dashed boxes

Proposition

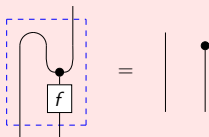
If $f : X \rightarrow Y$ has broad support, then



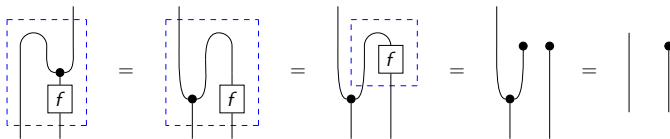
Disappearance of dashed boxes

Proposition

If $f : X \rightarrow Y$ has broad support, then



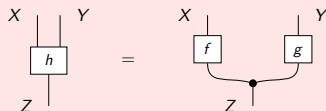
Proof:



Example: deriving conditional independence

Definition

Let $h : X \rightarrow Y \otimes Z$. We say that, in the context of h , Y is **conditionally independent** of Z given X , if h can be factorised as



In this case, we write $Y \perp\!\!\!\perp Z \mid X$.

Example: deriving conditional independence

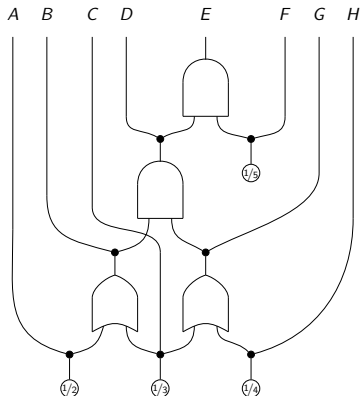
Prove that $A \perp\!\!\!\perp B \mid Z$ in the following program:

```
Z <- flip(1/2)
X <- flip(1/2)
Y <- flip(1/2)
A = X || Z
B = Z || Y
return (Z, X, Y, A, B)
```

Example: deriving conditional independence

Example: inference in a Bayesian network

Compute how conditioning on B effects E in the following fault tree:



Example: inference in a Bayesian network

Conclusion and outlook

- We developed a powerful compositional calculus for computing disintegrations in discrete probabilistic models.
- We recognised the role of Bayesian inversion and the notion of broad support.
- Many more examples in the paper:
 - ▶ Applications to probabilistic programming.
 - ▶ We give an elegant derivation of the ‘front-door-adjustment’ formula, a known example from the causal reasoning literature.
 - ▶ An example of counterfactual reasoning.

Conclusion and outlook

- We developed a powerful compositional calculus for computing disintegrations in discrete probabilistic models.
- We recognised the role of Bayesian inversion and the notion of broad support.
- Many more examples in the paper:
 - ▶ Applications to probabilistic programming.
 - ▶ We give an elegant derivation of the ‘front-door-adjustment’ formula, a known example from the causal reasoning literature.
 - ▶ An example of counterfactual reasoning.
- Question: how to generalise to continuous probability?
- Outlook: implementation in a string diagram rewrite tool.