Disintegration

Radboud University Nijmegen

Bart Jacobs — joint work with Kenta Cho
bart@cs.ru.nl
June 13, 2017

Where we are, so far

iCIS | Digital Security
Radboud University

Larger picture: channel-based probability theory

- Instances of channels are:
  - Kleisli maps for probability monads: $\mathcal{D}$, $\mathcal{G}$, ...
  - stochastic kernels
  - stochastic matrices, Markov chains
  - conditional probabilities $P(y \mid x)$
  - completely positive unital operators (in quantum)
- General underlying idea: maps in a (monoidal) effectus
  - states are channels $\omega \colon 1 \to X$, predicates are channels $p \colon X \to 2$
  - validity $\omega \models p$ is the scalar $p \circ \omega \colon 1 \to 2$
  - with graphical language
- Tool support for probability calculations via EfProb
  - an embedded language in Python
  - uniformly for discrete, continuous, quantum probability
  - see Calco Tools 2017 paper, and `efprob.cs.ru.nl`

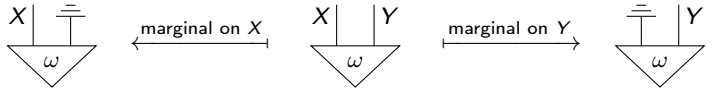
iCIS | Digital Security
Radboud University

Graphical language: channels as boxes (flow is upwards)

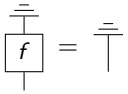
For symmetric monoidal categories with discarding (tensor unit is final):

- Sequential and parallel composition:

- States are triangles $1 \to X$, that can be marginalised via discarding $\top$



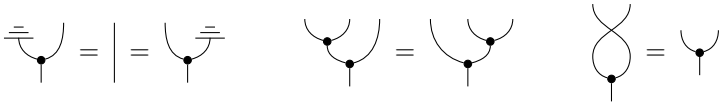
- By finality, channels are causal (or unital):



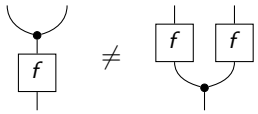
iCIS | Digital Security
Radboud University

Copying in classical probability

- In classical (but not quantum) probability one has copiers with:



- These copiers are not natural, in general, that is:



When f does satisfy this equation it is called deterministic.

Main examples categories

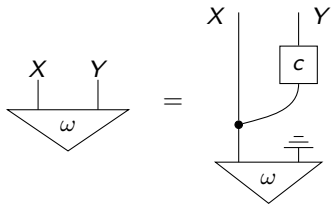
- $\mathcal{K}\ell(\mathcal{D})$, the Kleisli category of the (finite, discrete) distribution monad $\mathcal{D}$ on **Sets**
- $\mathcal{K}\ell(\mathcal{G})$, the Kleisli category of the Giry monad $\mathcal{G}$, on the category of measurable spaces
  - or on restrictions thereof, like Polish spaces, or standard Borel spaces
- $\mathcal{K}\ell(\mathcal{P}_+)$ the Kleisli category of the non-empty powerset monad on **Sets**.
- • • •

Where we are, so far

Disintegration definition

Disintegration is an operation that applies to joint states. It extracts a channel from the joint state, such that the original state can be reconstructed via this channel, copying and marginalisation.

A disintegration of a state ω on X, Y is a channel $c\colon X \to Y$ with:



- Disintegrations may not exist, and if they exist, they may not be determined uniquely.
- The probabilistic formulation is: $\omega(x, y) = \omega(y \mid x) \cdot \omega(x)$

What is disintegration good for?

- In a Bayesian network one efficiently represents a (multiple) joint state in graphical form
  - the graph structure captures conditional independence (not necessarily causality)
  - semantically, the arrows are channels, in $\mathcal{K}\ell(\mathcal{D})$
- Disintegration is the mechanism to go from the joint state to the arrows/channels in the graph
- In addition, Bayesian inversion is a special form of disintegration, see later
  - it can be used for belief updates, and for classification

In short, disintegration is a fundamental technique in probability

Disintegration in discrete probability

Let $\omega \in \mathcal{D}(X \times Y)$ be a joint state/distribution. One can extract a channel $c \colon X \to \mathcal{D}(Y)$ via:

$$c(x) = \sum_y \frac{\omega(x,y)}{\sum_{y'} \omega(x,y')} |y\rangle$$

This works only if the first marginal $\omega_X(x) = \sum_{y'} \omega(x,y)$ is nowhere zero. Then the disintegration (channel) is uniquely determined.

Disintegration in continuous probability

- For a joint state/distribution $\omega \in \mathcal{G}(X \times Y)$, a disintegration channel $c \colon X \to Y$ is often called a regular conditional probability.
- The defining disintegration equation becomes:

$$\omega(M \times N) = \int_M c(-)(N)\, \mathrm{d}\omega_X$$

wheren $\omega_X \in \mathcal{G}(X)$ is the X-marginal, with $\omega_X(M) = \omega(M \times Y)$.

Theorem (Faden 1985)

If Y is a standard Borel space (a compact metric space, with Borel σ-algebra), there is a unique disintegration $X \to Y$.

Disintegration for states given by pdf's

Let $\omega \in \mathcal{G}(X \times Y)$ be given by a pdf $u \colon X \times Y \to \mathbb{R}_{\geq 0}$, so that $\omega(M \times N) = \int_{M \times N} u(x,y)\, \mathrm{d}(x,y)$.

Theorem

The disintegration $c \colon X \to \mathcal{G}(Y)$ now is:

$$c(x)(N) = \int_N \frac{u(x,y)}{\int u(x,y')\, \mathrm{d}y'}\, \mathrm{d}y$$

This disintegration formula is implemented in EfProb, together with the very similar discrete version.

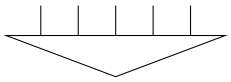
Where we are, so far

Marginalisation via masks

- So far we looked at binary joint states $1 \to X \otimes Y$
- More generally, we need n-ary states $1 \to X_1 \otimes \cdots \otimes X_n$
- There are then many ways to marginalise, namely 2^n

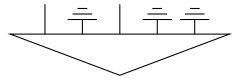
We use 0/1-masks, as follows. For

ω =



we write:

$\omega \,\%\, [1,0,1,0,0]$ =

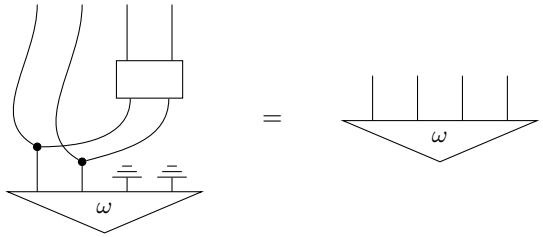


(This is actually EfProb notation)

iCIS | Digital Security
Radboud University

Disintegration via masks

There are also multiple ways to disintegrate an n-ary state.
For instance, one way to disintegrate a 4-ary state ω is:



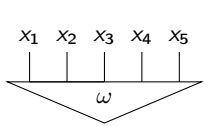
In that case we write the extracted box (channel) as:

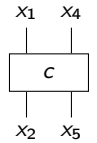
$$\omega \,/\!/\, [1,1,0,0]$$

iCIS | Digital Security
Radboud University

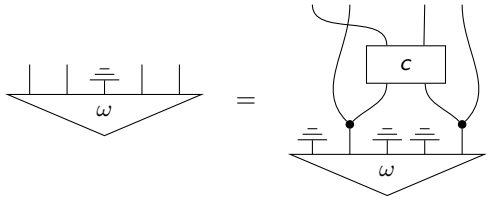
Combining marginalisation and disintegration

Consider a 5-ary state $\omega = \omega(x_1, x_2, x_3, x_4, x_5)$ for which we want to form the condiational probability $c = \omega(x_1, x_4 \mid x_2, x_5)$.





This channel c must satisfy:



We take: $c = \omega \,\%\, [1,1,0,1,1] \,/\!/\, [0,1,0,1] =: \omega\big[[1,0,0,1,0] : [0,1,0,0,1]\big]$

iCIS | Digital Security
Radboud University

Summary, so far

For an n-ary state/distribution ω, the conditional notation

$$\omega(\vec{x} \mid \vec{y}) \qquad \text{with } \vec{x}, \vec{y} \text{ disjoint}$$

has been formalised, in EfProb notation, with two masks:

$$\omega\big[\underbrace{[0/1, \ldots, 0/1]}_{1 \text{ for } x_i} : \underbrace{[0/1, \ldots, 0/1]}_{1 \text{ for } y_j}\big]$$

This is a powerful construct.

Where we are, so far

iCIS | Digital Security
Radboud University

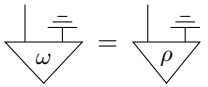
What is the natural join?

- Imagine two databases with student data, each given by a sequence of tuples, forming a relation — ie. a Kleisli map for powerset
  - The first contains student (registration) numbers and addresses
  - The second contains student numbers and exam marks
- If the 'marginals' of both databases/relations given by the student numbers coincide, then one can merge the two databases into one
  - the result is written with $\bowtie$ as $D_1 \bowtie D_2$
- We show how to do this via disintegration
  - the database example is in $\mathcal{K}\ell(\mathcal{P}_+)$, but the solution is much more general
  - the $\bowtie$ construction also came up in the contextuality work of Abramsky et al, on patching perspectives together

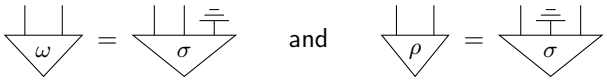
iCIS | Digital Security
Radboud University

Natural join: the question

Given two states ω and ρ with the same marginal:



is there a ternary state $\sigma = \omega \bowtie \rho$ that marginalises both to ω and to ρ:



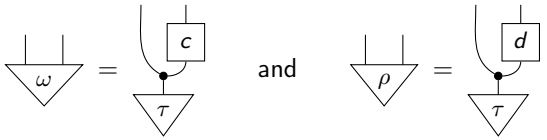
That is, $\omega = \sigma \,\%\, [1, 1, 0]$ and $\rho = \sigma \,\%\, [1, 0, 1]$.

iCIS | Digital Security
Radboud University

Natural join: the solution

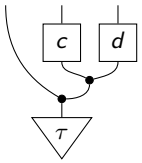
Let ω and ρ have the same first marginal, written as τ in:

Take disintegrations $c = \omega /\!\!/ [1,0]$ and $d = \rho /\!\!/ [1,0]$ of these states, so:



Then take as natural join:

$\omega \bowtie \rho \quad :=$



Probabilistic natural join example

- Consider sets $A = \{a_1, a_2\}$, $B = \{b_1, b_2, b_3\}$ and $C = \{c_1, c_2, c_3\}$
- With two joint distributions:

$$\omega = \tfrac{1}{4}|\, a_1 b_1 \,\rangle + \tfrac{1}{4}|\, a_1 b_2 \,\rangle + \tfrac{1}{6}|\, a_2 b_1 \,\rangle + \tfrac{1}{6}|\, a_2 b_2 \,\rangle + \tfrac{1}{6}|\, a_2 b_3 \,\rangle$$
$$\rho = \tfrac{1}{12}|\, a_1 c_1 \,\rangle + \tfrac{1}{3}|\, a_1 c_2 \,\rangle + \tfrac{1}{12}|\, a_1 c_3 \,\rangle + \tfrac{1}{8}|\, a_1 c_1 \,\rangle + \tfrac{1}{8}|\, a_1 c_2 \,\rangle + \tfrac{1}{4}|\, a_1 c_3 \,\rangle$$

- Their first marginals are equal, namely: $\tfrac{1}{2}|\, a_1 \,\rangle + \tfrac{1}{2}|\, a_2 \,\rangle$
- We represent these states as `w` and `r` in EfProb and compute their natural join

Probabilistic natural join example, continued

In EfProb we take the disintegrations `c`, `d` and then form the state `s`

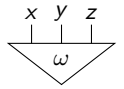
```
>>> c = w // [1,0]
>>> d = r // [1,0]
>>> t = w % [1,0]
>>> s = (idn(A) @ c @ d) * copy(A,3) >> t
```

This `s` is a big solution, consisting of $18 = 2 \cdot 3 \cdot 3$ terms. It has appropriate marginals equal to `w` and `r`.

```
>>> s % [1,1,0]
0.25|a1,b1> + 0.25|a1,b2> + 0|a1,b3> +
... 0.167|a2,b1> + 0.167|a2,b2> + 0.167|a2,b3>
>>> s % [1,0,1]
0.0833|a1,c1> + 0.333|a1,c2> + 0.0833|a1,c3> +
... 0.125|a2,c1> + 0.125|a2,c2> + 0.25|a2,c3>
```


Conditional independence: setting

Consider a ternary state ω with wires labeled x, y, z as in:



We wish to express the conditional independence of x, y given z

Commonly this conditional independence is written as:

$$\omega(x, y \mid z) = \omega(x \mid z) \cdot \omega(y \mid z)$$

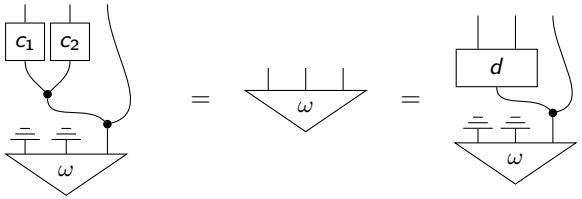
This can be expressed in terms of extracted channels, via disintegration

Expressing $\omega(x, y \mid z) = \omega(x \mid z) \cdot \omega(y \mid z)$

- LHS is the disintegration $d = \omega /\!\!/ [0, 0, 1]$
- RHS is given by two channels:

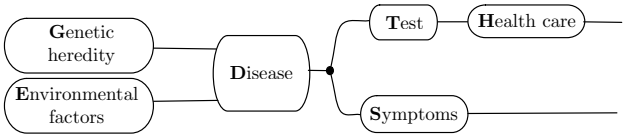
$$c_1 = \omega\big[[1, 0, 0] : [0, 0, 1]\big] \qquad c_2 = \omega\big[[0, 1, 0] : [0, 0, 1]\big]$$

Conditional independence then amounts to the first equation:



Bayesian networks

Consider a Bayesian network graph of the following form.

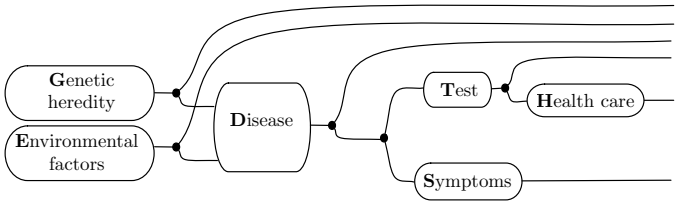


Each node with n parents corresponds to a channel $2^n \to \mathcal{D}(2)$, commonly described as a conditional probability table (cpt)

Bayesian network $\leftrightarrow$ Joint state + graph

A Bayesian network can be turned into a joint state, in two steps:

(1) one first makes all internal nodes 'externally visibile', by adding a link, as in:



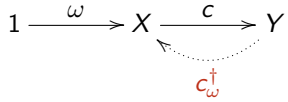
(2) One then computes the resulting state, by state transformation

In the other direction,

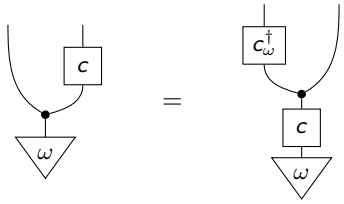
- combined marginalisation+disintegration gives the appropriate channels (cpt's).
- Ongoing work with Zanasi & Gaobardi; EfProb does the extractions

Bayesian inversion via decomposition

Bayesian inversion (Clerc et al 2017) turns a state and a channel into an inverted channel, written $c^\dagger_\omega$ in:



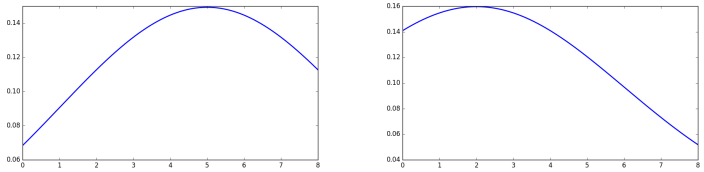
Graphically, disintegration is applied to the diagram on the left, giving the defining equation for Bayesian inversion $c^\dagger_\omega$ in:



Example: customers calling

Imagine a call centre that is open for 8 hours on each day of the week. The distribution of calls is different on weekends (Sat-Sun) from other days (Mon-Fri).

The probability density functions of the two distributions on $[0, 8]$ are:



What can we then learn from a single call at a given time of the day regarding whether it is weekend or not?

- Technically, this is a point observation, which is problematic in continuous conditioning

Customers calling: Bayesian inversion (in EfProb)

We translate the situation into a channel $2 \to [0, 8]$, with a prior on 2

The updated weekend/weekday probability after a call at **6** is:

```
>>> prior
0.286|W> + 0.714|~W>
>>> c = chan_from_states([gaussian_state(5,4,R(0,8)),
...                       gaussian_state(2,4,R(0,8))], ..)
>>> d = c.inversion(prior)
>>> d(6)
0.374|W> + 0.626|~W>
```

Recall/note:

- the EfProb implementation uses that channels are given by pdf's (likelihoods)
- the example is in fact a hybrid Bayesian network: it involves both discrete and continuous probability

Example: naive Bayesian classification

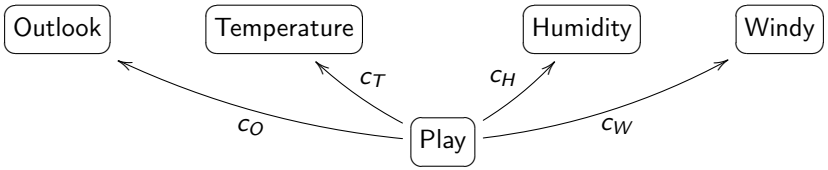
Consider the following table of data (Witten et al 2011)

Outlook	Temperature	Humidity	Windy	Play
Sunny	hot	high	false	no
Sunny	hot	high	true	no
Overcast	hot	high	false	yes
Rainy	mild	high	false	yes
Rainy	cool	normal	false	yes
Rainy	cool	normal	true	no
Overcast	cool	normal	true	yes
Sunny	mild	high	false	no
Sunny	cool	normal	false	yes
Rainy	mild	normal	false	yes
Sunny	mild	normal	true	yes
Overcast	mild	high	true	yes
Overcast	hot	normal	false	yes
Rainy	mild	high	true	no

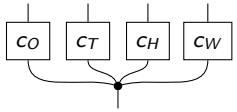
Question: Given a sunny outlook, cold temperature, high humidity and windiness. What is then the probability of playing?

From table to channels

We organise these data into a four channels c_O, c_T, c_H, c_W in a network:



The network is turned into a circuit with a single channel:



Its Bayesian inversion will be used for classification

In naive classification one assumes that all data are independent, given by separate channel. Despite this simplification, the method works remarkably well.

Classification via inversion

By counting occurrences we get four channel definitions:

```
>>> cO=chan_from_states([State([2/9,4/9,3/9],Outlook),
...                     State([3/5,0/5,2/5],Outlook)],Play)
>>> cT=chan_from_states([State([2/9, 4/9, 3/9],Temp),
...                      State([2/5, 2/5, 1/5],Temp)],Play)
>>> cH=chan_from_states([State([3/9, 6/9],Humidity),
...                     State([4/5, 1/5],Humidity)],Play)
>>> cW=chan_from_states([State([3/9, 6/9],Windy),
...                     State([3/5, 2/5],Windy)],Play)
```

These four channels are combined into one, which can be inverted:

```
>>> c = (cO @ cT @ cH @ cW) * copy(Play,4)
>>> c.inversion(prior_play)('Su', 'Co', 'Hi', 't')
0.205|y> + 0.795|n>
```

This same Play probability occurs in Witten et al. Our inversion approach also works for the continuous version of this example, with Gaussians

Where we are, so far

iCIS | Digital Security
Radboud University

Final remarks

- Channel-based probability offers a both abstract clarity and practical usefulness
- Disintegration (regular conditional probability) is a basic mathematical construction
  - it can be formulated abstractly in monoidal categories with copying and discarding
  - but it is mostly used in a probabilistic setting
- Here focus mostly on applications of disintegration
  - these applications require flexible formulation, using masks
  - other work focuses more on (unique) existence and properties
- Currently ongoing work: conjugate priors, "quantum disintegration", channel-based probabilistic programming language
- Today's work is unpublished, so far

iCIS | Digital Security
Radboud University