# Causal Deconvolution by Algorithmic Generative Models

Zenil et al., Nature Machine Intelligence 2019

MLRG Discussion, 05/FEB/2019

Lead by: Efrén Cruz Cortés

Obvi: New Journal - “**Nature Machine Intelligence**”

Question: what does this mean for free journals like JMLR and NeurIPS proceedings?

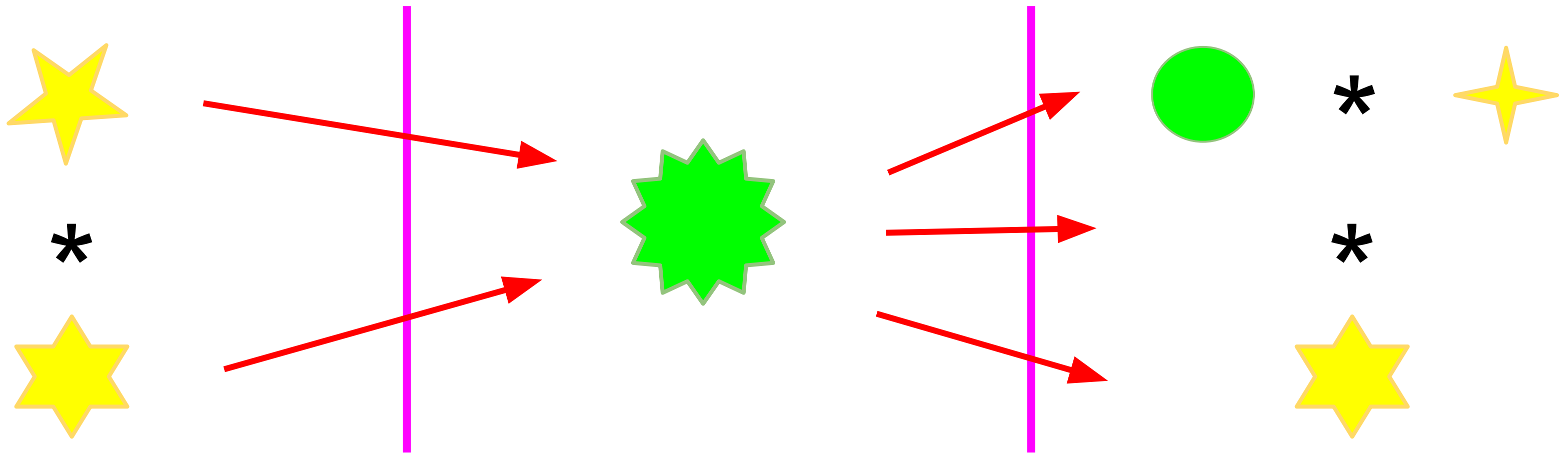
This Paper: “Let’s quickly put together all our previous research with no background”.

# Outline

1. Background
2. Paper Discussion
3. Connections / Applications Discussion

# Background

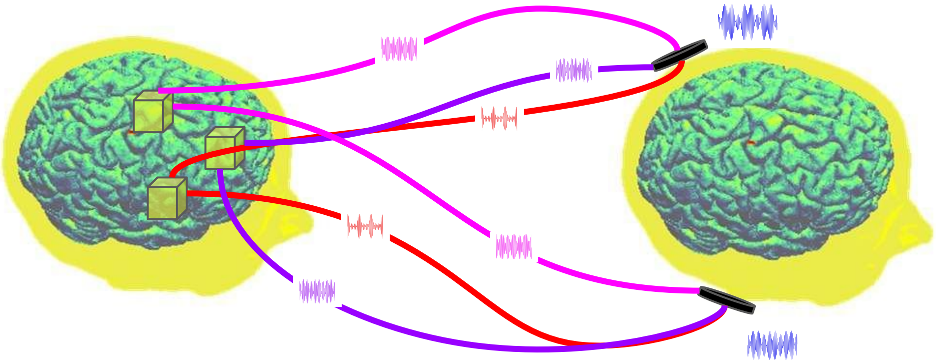
# The Deconvolution Problem




| Two or more elements are combined in some fashion | Observed Convolution | Deconvolution |
| --- | --- | --- |

**Note**: “Convolution” used loosely

# Examples

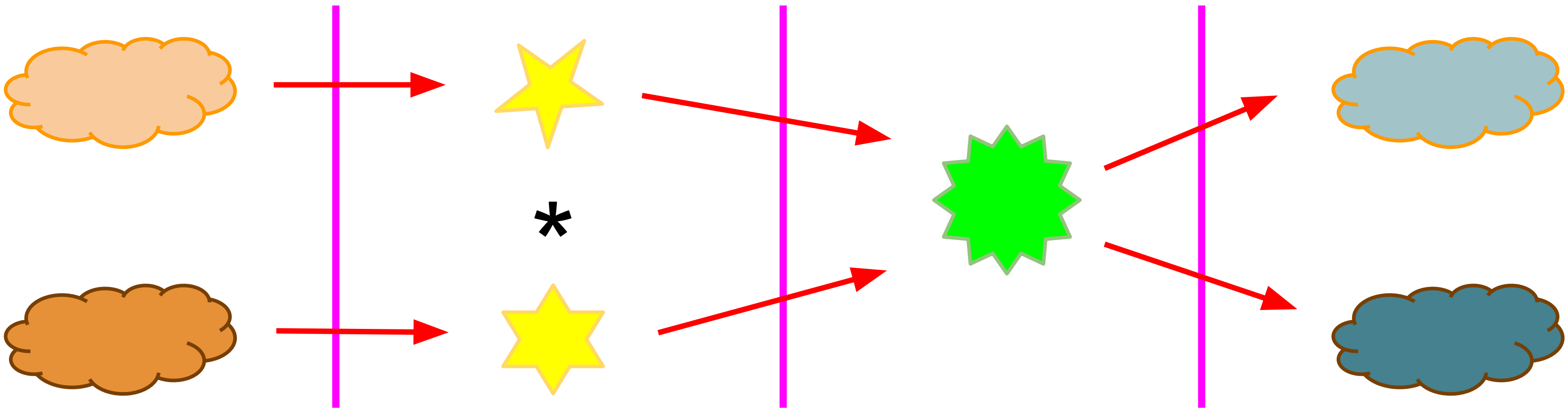


Brain Image Courtesy of Luis Gomez, Duke University

# Examples

- Independent Component Analysis
- Principal Component Analysis
- Fourier Analysis
- Network Community Detection
- Image Segmentation
- Audience

# Causal Deconvolution Problem




Generative Mechanisms

# Identifiability Problem


In general, deconvolution does not yield unique answers

**“Hat”**

Which is it?

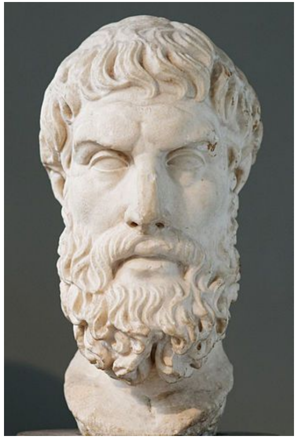
**“Elephant and Boa”**

Drawings by A. de S-E.

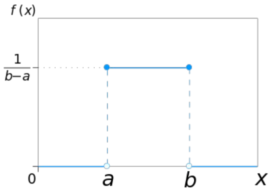
# Identifiability - Reworded

Given evidence in the form of data **D**, and a set of possible explanations, or hypotheses **H = {h}**, which one should we choose?

**"Principle of Multiple Explanations"**



**Keep them all! LOL**




Epicurus

**"Occam's razor"**

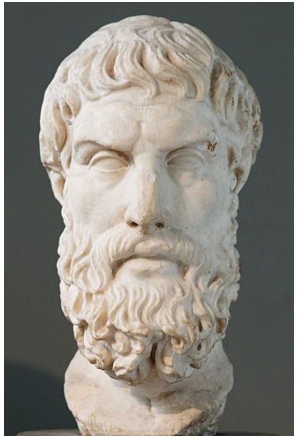
**Off with their heads! SMH**

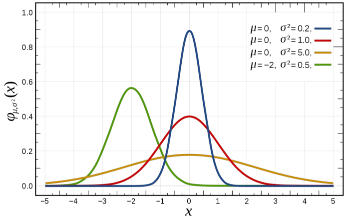
Occam

Images from Wikipedia

# Solomonoff Induction

- Give all feasible hypotheses non-zero prior probability.
- Give higher probability to “simpler” hypotheses.






Non-uniform Epicurus

Probabilistic Occam

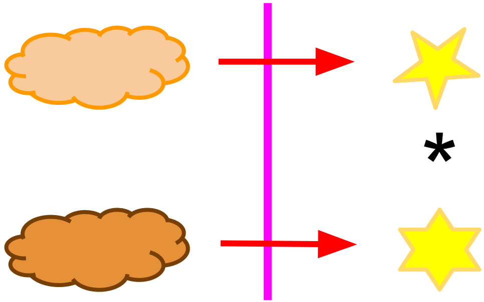
Images from Wikipedia

# Bayesian Connection

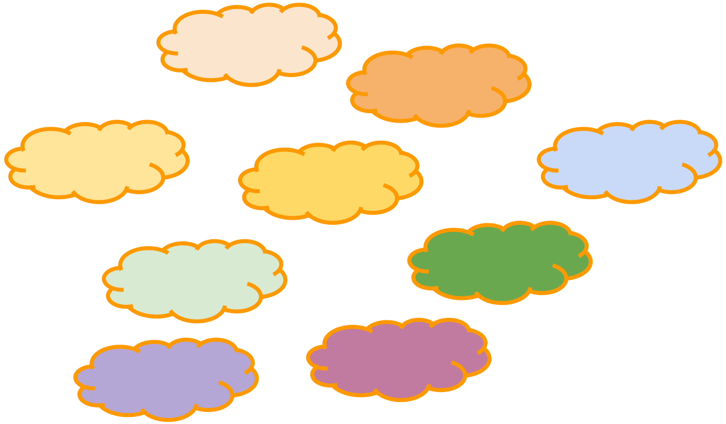
Solomonoff's Induction provides principle for choosing prior on hypotheses.

$$P(h \mid D) \sim P(D \mid h) P(h)$$

# Revisit Causal Deconvolution



**Generative Mechanisms**



**Hypothesis Space**

**Which is “simpler”?**

# Automata Theory Formulation

“The world is a set of strings”.

**Strings**: sequences of symbols.

Examples:

1. DNA sequences
2. Images - long sequence of pixel RGB values
3. Networks - Adjacency Matrix
4. Audience

# Computable Strings

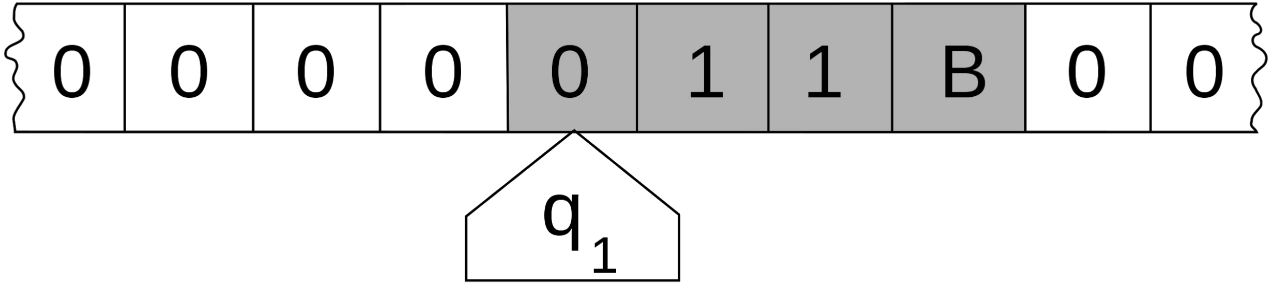
**But**, not all strings are part of the world.

Only those which are “computable” in the Turing sense:

They can be computed by some “algorithm” running on a **Turing Machine**.

What’s a Turing Machine?

# Turing Machine




A set of finite rules determines the dynamics of such machine

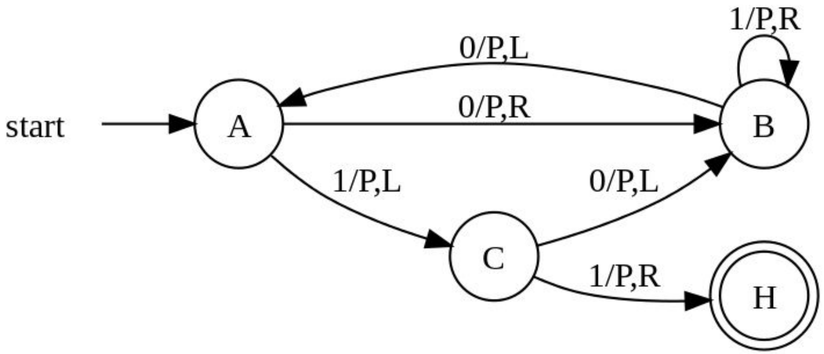
Images from Wikipedia

# More on Turing Machines

Do not be discouraged by simplicity, many complex “machines” turn out to be equivalent to Turing Machine.

**Universal Turing Machine**: Runs all Turing Machines.

# Example: Busy Beaver




- Three possible states
- Two Symbols

**The table for the 3-state busy beaver ("P" = print/write a "1")**

| Tape symbol | Current state A | | | Current state B | | | Current state C | | |
|---|---|---|---|---|---|---|---|---|---|
| | Write symbol | Move tape | Next state | Write symbol | Move tape | Next state | Write symbol | Move tape | Next state |
| **0** | P | R | **B** | P | L | **A** | P | L | **B** |
| **1** | P | L | **C** | P | R | **B** | P | R | **HALT** |

# Back to “Simple Strings” question

The complexity of a string is determined by the shortest “computer programs” that derive that string in a Universal Turing Machine.

Algorithmic Probability: The prior for a hypothesis **h** is:

$$\mathbf{P(h) \sim \sum 2^{-L(p)}}$$

Where summation is over all finite programs deriving **h,** and **L** is their length (for example, in bits, if **p** itself is a binary string).

# Complexity of a hypothesis

**P** is in not computable. But we can approximate: **f ~ P**

In paper, they derive **C(h)** from **f**.

# Paper Discussion

**Go to Paper Discussion**

# Applications / Connections ?

Note 1: separable Hilbert spaces are isomorphic to space of square-summable sequences. Can we formulate similar deconvolution algorithms in more familiar settings?

Note 2: Take same idea but frame it in terms of what “machines” we know (classifiers, etc.). What’s their complexity measure?

Note 3: Given complex phenomenon, use ABMs as hypotheses. What’s complexity measure?

**Audience**

# Applications / Connections ?

Note 4: similar to Statistical Learning Theory?

**h* = min_{h in H} [ L(h,D) + R(h) ]**

First term **L(h,D)** -> **h** can approximately “derive” **D**

Second term **R(h)**: **h** is “simple” (regularized)