Logic and Knowledge Representation

Representation, Complexity and Intelligence

April 3, 2017

Pierre-Alexandre Murena

TELECOM ParisTech

Introduction

What you have learned in the previous classes:

- Logic and deductive reasoning
- Knowledge representation, for example with semantic networks
- Natural language processing
- Inductive logic programming

All those problems are related to **Artificial Intelligence**.

Today, we will explore how they are related to a notion called *complexity*.

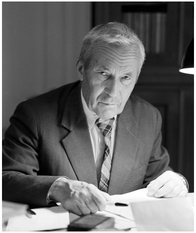
Overview

1. **Representation:** What is the optimal representation for a given object? Is it important to choose an optimal representation?

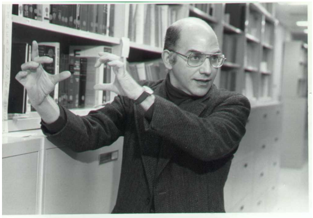
2. **Complexity:** Introduction of a core notion in artificial intelligence: *complexity*. Complexity is directly related to representation.

3. **Intelligence:** Why is complexity so helpful in artificial intelligence?

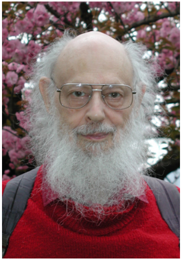
Starring



Andrei Kolmogorov



Gregory Chaitin

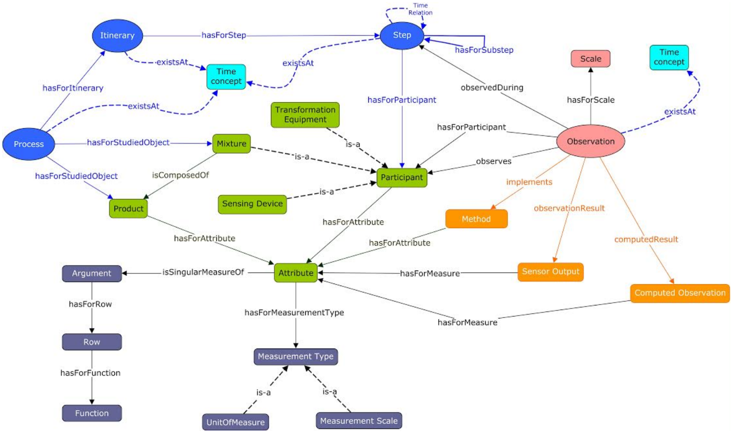


Ray Solomonoff

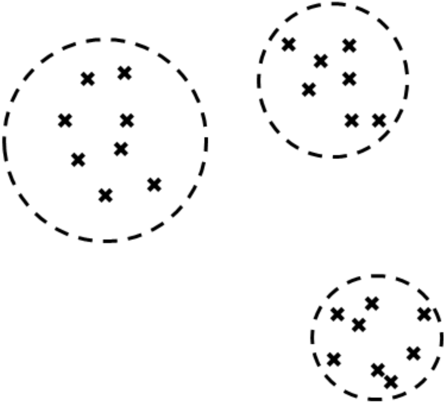
Outline

Outline

Semantic networks

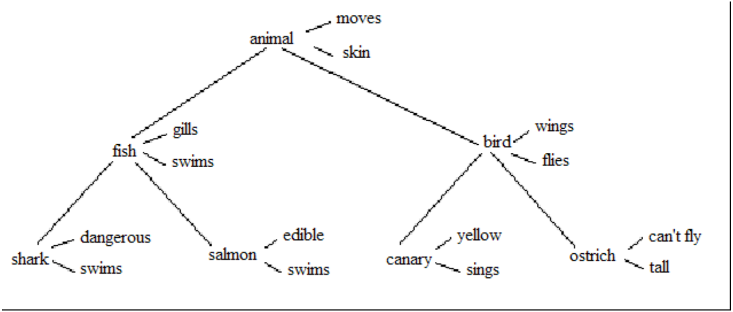


Clustering



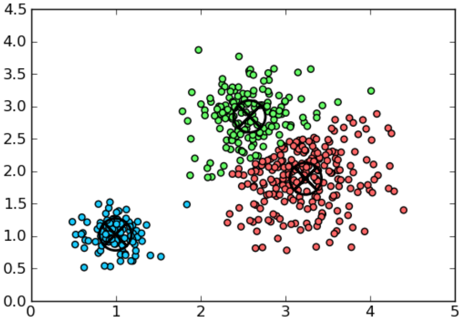
Outline

Semantic networks



Why do we choose to specify that birds fly and ostriches don't?

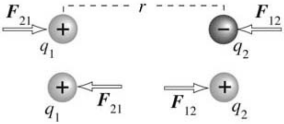
Clustering: K-means algorithm



Why is it useful to express relative positions toward cluster centers?

Analogy in representation

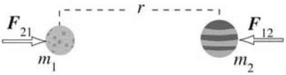
Electrostatic Force vs. Gravitational Force



$$F = k \frac{q_1 q_2}{r^2}$$

Electrostatic Force

F	=	electrostatic force
q	=	electric charge
r	=	distance between centers of charge
k	=	Coulomb constant 9.0×10^9 N · m²/C²



$$F = G \frac{m_1 m_2}{r^2}$$

Gravitational Force

F	=	gravitational force
m	=	mass
r	=	distance between centers of mass
G	=	gravitational constant 6.7×10^{-11} N · m²/kg²

Outline

Outline

Informal definition of Kolmogorov complexity

The complexity of an object corresponds to the minimal length of a computer program producing this object.

For example: The number $111\ldots111$ is not very complex:

```
for i = 1 .. n:
  print 1
```

Reminder: Turing machine

What is a Turing Machine?

Reminder: Turing machine

Digression

Reminder: Turing machine
Informal definition

Turing machine

A Turing machine corresponds to a sequential computer program which can be executed with an input, produces an output and uses an infinite memory.

Reminder: Turing machine

More formal definition

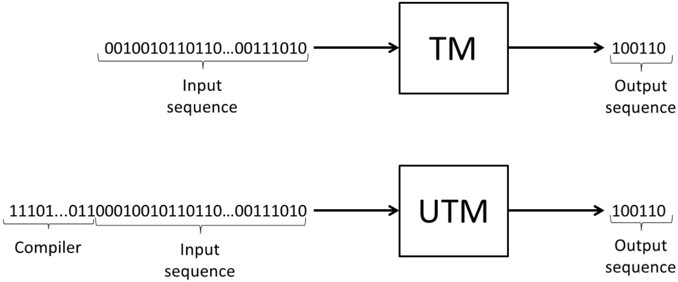
Definition

A Turing Machine is an automaton writing **symbols** (from an alphabet Σ) on an **infinite tape** (ie. a linear list of cells), using a **head** (ie. an access pointer).

- At each time step, the head scans the content of current cell
- Using an association function $(symbol, state) \mapsto action$, the device either writes a symbol or shifts to an adjacent cell.

Universal Turing machine

A **Universal Turing Machine** (UTM) is a Turing machine simulating any other Turing machine.



Defining complexity

Preliminary thought

Several programs can produce the same object x.

or

Several Turing machines can produce the same output x.

Question

Which program is the most adapted to describe the *complexity* of x?

Machine dependent complexity

The complexity of an object x relative to a UTM $\mathcal{M}$ is defined as the length of the shortest program on $\mathcal{M}$ producing object x.

$$C_{\mathcal{M}}(x) = \min_{p \in \mathcal{P}_{\mathcal{M}}} \{l(p) : p() = x\}$$

What is the problem with this definition?

Defining complexity

From one machine to another

Question

What can be said about the quantity $|C_{\mathcal{M}_2}(x) - C_{\mathcal{M}_1}(x)|$?

Defining complexity

From one machine to another

Question

What can be said about the quantity $|C_{\mathcal{M}_2}(x) - C_{\mathcal{M}_1}(x)|$?

Invariance theorem

There exists a constant $c_{\mathcal{M}_1,\mathcal{M}_2}$ such that for any object x:

$$|C_{\mathcal{M}_2}(x) - C_{\mathcal{M}_1}(x)| < c_{\mathcal{M}_1,\mathcal{M}_2}$$

Defining complexity
Machine independent complexity

Consequence of the Invariance theorem: The complexity of an object is an intrinsic property of the object, which does not depend on the machine.

Example

Is the number π complex?

Example

Is the number π complex?

No!

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$$

Back to the invariance theorem

The constant $c_{\mathcal{M}_1,\mathcal{M}_2}$ can be as large as the considered objects!

Example

There exists a machine $\mathcal{M}$ on which my PhD thesis is entirely written. Accessing my thesis on this machine is done with the program:

```
if p[0] == 0 :    print(PhD_thesis_content)
else:    return M(p[1:])
```

Hence: $C_{\mathcal{M}}(\text{My PhD thesis}) = 1$

Invariance theorem (again)

Proposition

For all UTM $\mathcal{M}$ and for all object x, we have:

$$C(x) \leq C_{\mathcal{M}}(x) + C(\mathcal{M})$$

Outline

Incomputability

Theorem

Kolmogorov complexity is incomputable.

Conditional complexity

Definition: Conditional complexity

Given a machine $\mathcal{M}$, the complexity of object x knowing object y is defined as:

$$C_{\mathcal{M}}(x|y) = \min_{p \in \mathcal{P}_{\mathcal{M}}} \{l(p) : p(y) = x\}$$

Properties

Theorem

There is a constant c such that for all x and y, $C(x) \leq l(x) + c$ and $C(x|y) \leq C(x) + c$

Properties

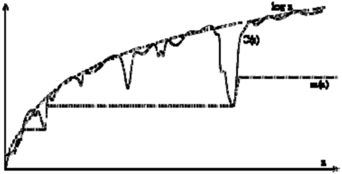
Chain rule

$$C(x) \leq C(y) + C(x|y)$$

Lemma

$$C(x, y) \leq C(x) + C(y|x)$$

C as an Integer Function

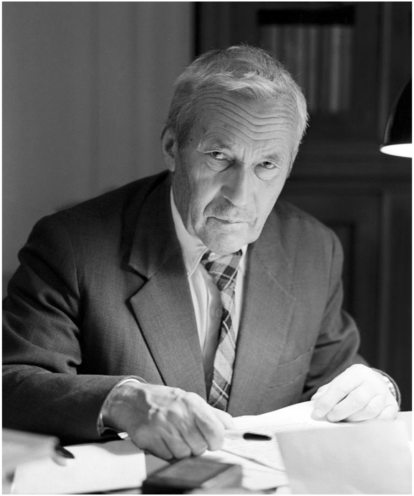


Theorem

1. The function $x \mapsto C(x)$ is unbounded
2. The function $m(x) = \min\{C(y) : y > x\}$ is unbounded.
3. The function $C(x)$ is continuous: there is a constant c such that $|C(x) - C(x+h)|| \leq 2l(h) + c$
4. The function $C(x)$ mostly "hugs" $\log x$: $C(x) \leq \log x + c$

Outline

Randomness



Randomness


TOUR OF ACCOUNTING
OVER HERE WE HAVE OUR RANDOM NUMBER GENERATOR.
NINE NINE NINE NINE NINE NINE
ARE YOU SURE THAT'S RANDOM?
THAT'S THE PROBLEM WITH RAN- DOMNESS: YOU CAN NEVER BE SURE.
10/25/01 © 2001 United Feature Syndicate, Inc.

Randomness

Claim: I have a true fair coin.
Challenge: Let's flip it 26 times and check that!

1. 00000000000000000000000000
2. 01000110110000010100111001
3. 10010011011000111010110010

In which scenario is my claim true?

Randomness

Claim: I have a true fair coin.
Challenge: Let's flip it 26 times and check that!

1. $\Pr(00000000000000000000000000) = 1/2^{26}$
2. $\Pr(01000110110000010100111001) = 1/2^{26}$
3. $\Pr(10010011011000111010110010) = 1/2^{26}$

The three sequences have equal chances to be observed!

Randomness

Okay... But probability is about frequency... We expect the 0s and 1s to appear at the same rate!

Randomness

1. 00000000000001111111111111
2. 10010011011000111010110010

Are both sequences equally random?

Randomness

A definition

A finite sequence is said to be *random* if it is incompressible, ie. if its shortest description is the sequence itself.

This definition only works for **finite** sequences.

Randomness

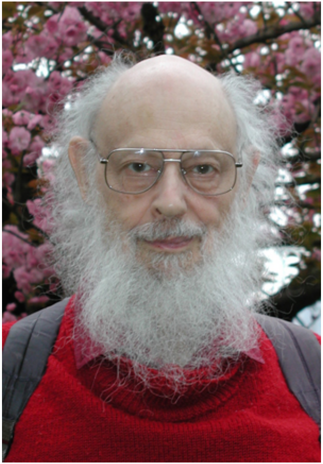
Do random sequences exist?

- Consider binary sequences of length L. There exists 2^L such sequences.
- A proportion 2^{-k} of them can be compressed to k bits exactly.
- Number of compressible sequences:

$$1 + 2 + 2^2 + 2^{L-1} = 2^L - 1$$

Conclusion: Some sequences cannot be compressed.

Probabilities



Probabilities

Motivation: Assign a *universal* probability to each finite binary string.

Probabilities

Motivation: Assign a *universal* probability to each finite binary string.

First attempt

$$P(x) = \sum_p 2^{-l(p)}$$

P does not define a proper probability density function:

- $\sum_x P(x) = \infty$
- $P(x) = \infty$ for all x

Probabilities

Motivation: Assign a *universal* probability to each finite binary string.

Second attempt

$$P(x) = 2^{-C(x)}$$

P does not define a proper probability density function: $\sum_x P(x) = \infty$

Definition: Prefix codes

In a prefix code, no *code word* is the prefix to another code word.

Why is this property useful in coding theory?

Definition: Prefix complexity

The prefix complexity of x (denoted $K(x)$) is the size of the shortest self-delimited program that outputs x when run on a given universal Turing machine $\mathcal{M}$.

$$K_{\mathcal{M}}(x) = \min_{p \in \mathcal{PP}_{\mathcal{M}}} \{l(p) : p() = x\}$$

Remark: $C(x) \leq K(x) \leq C(x) + 2\log C(x) + \mathcal{O}(1)$

Definition: Universal Distribution

$$m(x) = \sum_{p \in \mathcal{P}\mathcal{P}_{\mathcal{M}}} 2^{-l(p)}$$

Property

$m(x) = \mathcal{O}\left(2^{-K(x)}\right)$ or equivalently $-\log m(x) = K(x) + \mathcal{O}(1)$

Remark: Why is it called *universal*?

Outline

Outline

Analysis of deduction

Deduction examples (1)

1. All men are mortal.
2. Plato is a man.
3. Therefore, Plato is mortal.

Analysis of deduction

Deduction examples (2)

Cauchy-Schwarz inequality

Let $\alpha = (a_1, \ldots, a_n)$ and $\beta = (b_1, \ldots, b_n)$ be two sequences of real numbers. Then:

$$\left(\sum_{i=1}^{n} a_i^2\right)\left(\sum_{i=1}^{n} b_i^2\right) \geq \left(\sum_{i=1}^{n} a_i b_i\right)^2$$

Proof

Analysis of deduction

Deduction examples (2)

Cauchy-Schwarz inequality

Let $\alpha = (a_1, \ldots, a_n)$ and $\beta = (b_1, \ldots, b_n)$ be two sequences of real numbers. Then:

$$\left(\sum_{i=1}^{n} a_i^2\right)\left(\sum_{i=1}^{n} b_i^2\right) \geq \left(\sum_{i=1}^{n} a_i b_i\right)^2$$

Proof

For any $t \in \mathbb{R}$:

$$0 \leq \|\alpha + t\beta\|^2 = \|\alpha\|^2 + 2\langle\alpha, \beta\rangle t + \|\beta\|^2 t^2 = P(t)$$

The quadratic polynomial P is positive, so its discriminant is negative:

$$4|\langle\alpha, \beta\rangle|^2 - 4\|\alpha\|^2\|\beta\|^2 \leq 0$$

A definition for deductive reasoning

Deductive reasoning is an approach where a set of logic rules are applied to general axioms in order to find (or more precisely *to infer*) conclusions of no greater generality than the premises.

A definition for deductive reasoning

Deductive reasoning is an approach where a set of logic rules are applied to general axioms in order to find (or more precisely *to infer*) conclusions of no greater generality than the premises.

Or, less formally:

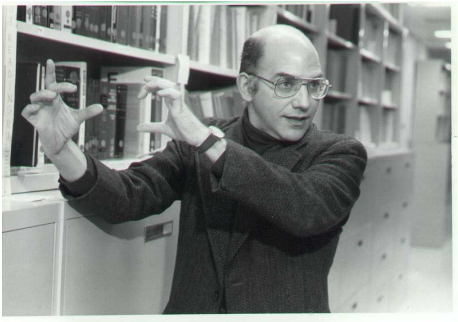
- General $\longrightarrow$ Less general
- General $\longrightarrow$ Particular

Deduction and compression

Godel's first incompleteness theorem

Any consistent formal system F within which a certain amount of elementary arithmetic can be carried out is incomplete; i.e., there are statements of the language of F which can neither be proved nor disproved in F.

Deduction and compression



Intuition: A formal system is a compression of the set of theorems it can prove. *Thus, there is an intrinsic limit to what the system can do.*

Deduction and compression

Proving randomness... or not!

Chaitin's theorem

Statements such as "$K(n) > m$" cannot be proven above a certain value of m, though they are true for infinitely many integers n.

In particular: Although most strings are random, it is impossible to effectively prove them random.

Outline

Limits of deduction

Will it rain today?

Limits of deduction

We are hardly able to get through one waking hour without facing some situation (e.g. *will it rain or won't it?*) where **we do not have enough information** to permit deductive reasoning; but still we must decide immediately.
In spite of its familiarity, the formation of plausible conclusions is a very subtle process.

in [Edwin T. Jaynes, *Probability theory. The logic of science*, Cambridge U. Press, 2003]

Examples of conclusions of non-deductive reasoning

- It will rain today.
- All dogs bark.
- Everybody in this room knows that $1 + 1 = 2$
- The sun always rises in the East.
- Life is not a dream.
- ...

Inductive reasoning

Definition

Inductive reasoning is an approach in which the premises provide **a strong evidence** for the truth of the conclusion.

The conclusion of induction is not guaranteed to be true!

A frequent confusion

Deduction: General rule $\Longrightarrow$ Particular case
Induction: Particular case $\Longrightarrow$ General rule

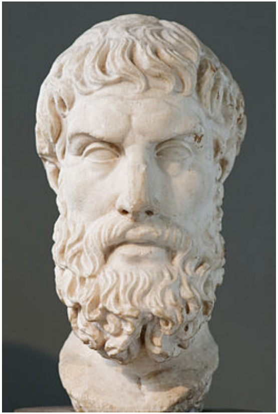
A frequent confusion

Deduction: General rule $\Longrightarrow$ Particular case
Induction: Particular case $\Longrightarrow$ General rule

This is incorrect!

Philosophical treatment

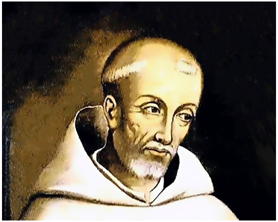
Epicurus (342-270 B.C.)



Principle of Multiple Explanations: If more than one theory is consistent with the observations, keep all theories.

Philosophical treatment

William of Ockham (1290-1349)



Occam's Razor Principle: Entities should not be multiplied beyond necessity

Philosophical treatment

Thomas Bayes (1702-1761)

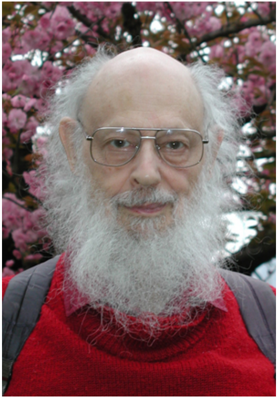
Probabilistic point of view on inductive reasoning.

Bayes's Rule: The probability of hypothesis H being true is proportional to the learner's initial belief in H (the *prior probability*) multiplied by the conditional probability of D given H.

A fundamental question

What is the justification for inductive reasoning?

General principle



Solomonoff's Lightsaber

Combining the **Principle of Multiple Explanations**, the **Principle of Occam's Razor**, **Bayes Rule**, using **Turing Machines** to represent hypotheses and **Algorithmic Information Theory** to calculate their probability.

Principle of Multiple Explanations

All hypotheses explaining the data have to be considered.

Only the hypotheses discarded by the data can be rejected.

Even if all hypotheses are considered, the most complex hypotheses must be dropped when we find simpler ones.
This idea is basically derived from Occam's Razor.

Step 3: Bayes Rule

To neglect complex hypotheses, Bayes rule can be used with high priors for simple hypotheses and low priors for complex hypothes:

$$Pr(H_i|D) = \frac{Pr(D|H_i) \times Pr(H_i)}{Pr(D)}$$

where the value of $Pr(H_i)$ is low if H_i is complex and high if H_i is simple.

Solomonoff's approach step by step

Step 4: Encoding hypotheses with Universal Turing Machines

- Data D are encoded as a sequence over a finite alphabet $\mathcal{A}$ (for example binary alphabet $\mathcal{A} = \{0, 1\}$).
- Hypotheses are processes: hence, they can be represented as Turing Machines (TM).
- Hypotheses are represented as input sequences of Universal Turing Machines (UTM).
- The set of possible inputs of a UTM corresponds to the set of hypotheses.

Solomonoff's approach step by step

Step 5: Universal prior

The priors are chosen to be:

$$Pr(H_i) = m(H_i) \simeq 2^{-K(H_i)}$$

Solomonoff's Induction

1. Run any possible hypothesis H_i on the UTM:
   - If H_i produces the data D:
     1.1 Accept the hypothesis: $Pr(D|H_i) = 1$
     1.2 Calculate Kolmogorov complexity of H_i: $K(H_i)$
     1.3 $Pr(H_i) = m(H_i) \simeq 2^{-K(H_i)}$
   - Otherwise: Discard the hypothesis: $Pr(D|H_i) = 0$
2. $H^* = \arg\max_{H_i}\{Pr(H_i) \times Pr(D|H_i)\}$

This problem is intractable!

So what?

The strongest result of this theory is that **a universal distribution can be used as an estimator *for all priors***.

So what?

The strongest result of this theory is that **a universal distribution can be used as an estimator *for all priors*.**

Theorem

If μ is the *concept* computable measure and the conditional semi-measure $\mu(y|x)$ is defined by $\mu(y|x) = \frac{\mu(xy)}{\mu(x)}$.
Let $\mathcal{B}$ be a finite alphabet and x a word over $\mathcal{B}$. The summed expected squared error at the n-th prediction is defined by:

$$S_n = \sum_{a \in \mathcal{B}} \sum_{l(x)=n-1} \mu(x) \left(\sqrt{\mathbf{M}(a|x)} - \sqrt{\mu(a|x)} \right)^2$$

Then $\sum_n S_n \leq K(\mu) \log(2)$

Outline

Induction and bias

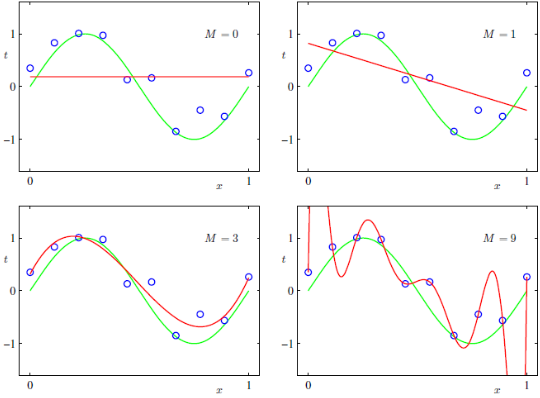
Remarks

1. An inductive algorithm is **biased** toward a given class of problems.
2. The performance of an algorithm is **necessarily** relative to a class of problems.
3. Induction does not create information: it only *transforms* a prior information contained in the algorithm.

Two classes of bias

1. **Representation bias**: a bias on the form of the concept
2. **Research bias**: a bias on how the concept is searched

Example: regression



Which model would you choose?

Inductive principle
Bayes rule

$$\hat{H} = \arg\max_{H_i} \quad \frac{Pr(D|H_i) \times Pr(H_i)}{Pr(D)}$$

Minimum Description Length Principle

The best theory to describe observed data is the one which minimizes the sum of the description length (in bits) of:

- the theory description
- the data encoded from the theory

Inductive principle
Minimum Description Length Principle

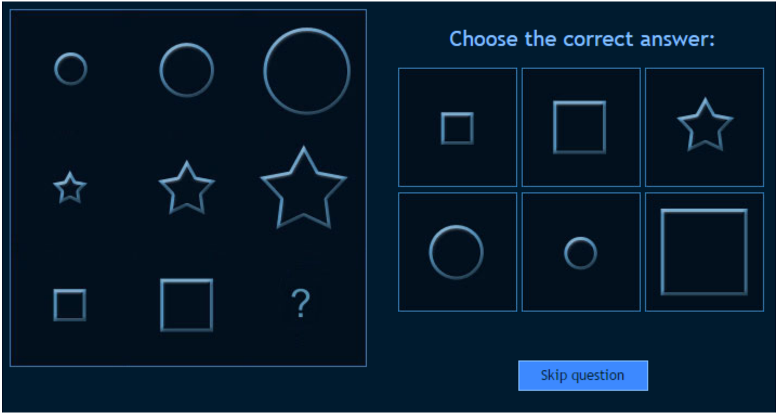
$$\hat{H} = \arg\min_{H_i} \quad C(H_i) + C(D|H_i)$$

or

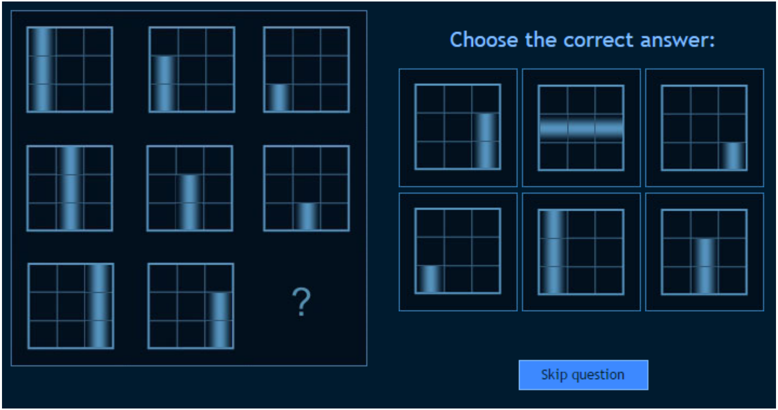
$$\hat{H} = \arg\min_{H_i} \quad K(H_i) + K(D|H_i)$$

Outline

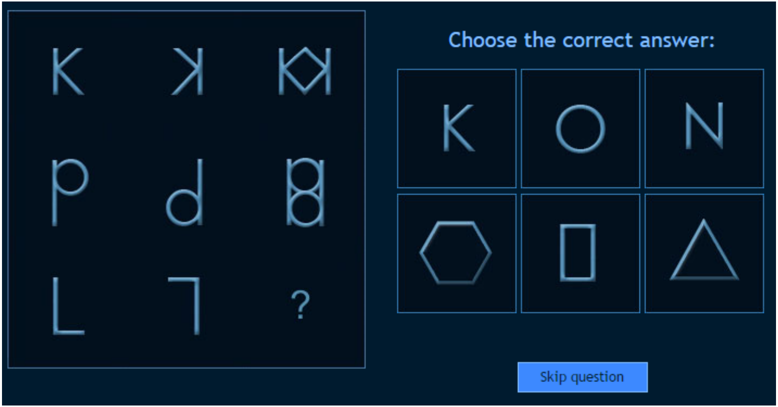
IQ tests



IQ tests



IQ tests



What to say about these problems?

- Inductive problems
- Repetition of *similar* structures
- A question is asked about a missing state
- Search of regularity

What to say about these problems?

- Inductive problems
- Repetition of *similar* structures
- A question is asked about a missing state
- Search of regularity

Such a situation is called an analogy

Analogy Reasoning

Definition (Analogy reasoning)

Analogy reasoning is a form of reasoning in which one entity is inferred to be similar to another entity in a certain respect, on the basis of the known similarity between the entities in other respects.

Analogy Reasoning

Definition (Analogy reasoning)

Analogy reasoning is a form of reasoning in which one entity is inferred to be similar to another entity in a certain respect, on the basis of the known similarity between the entities in other respects.

Definition (Proportional Analogy)

Proportional Analogy concerns any situation of the form “A is to B as C is to D”

Notation

$$A : B :: C : D$$

Examples

Analogy by Rendition

Occam's razor / Solomonoff's lightsaber

Works because of the underlying concept of *inductive principle*

Examples

Proportional analogy

- Gills are to fish as lungs are to man.
- François Hollande is to France as Vladimir Putin is to Russia
- Donald Trump is to Barack Obama as Barack Obama is to George Bush
- 37 is to 74 as 21 is to 42
- The sun is to Earth as the nucleus is to the electron

Analogy equation

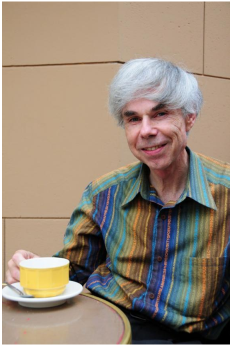
Definition (Analogy equation)

D is a solution of the analogy equation $A : B :: C : x$ iff $A : B :: C : D$

Remarks on analogy equation

- Solving an analogy equation is a typical inductive reasoning problem.
- Several solutions may be equally correct for an equation
- The quality of a solution is dependent of the machine.

Douglas Hofstadter (1945-now)



"We are trying to put labels on things by mapping situations that we have encountered before. That to me is nothing but analogy."

A micro-world

- Alphabet $\Sigma = \{A, B, C, \ldots, Z\}$
- Elements of the analogy are words over Σ

A micro-world

- Alphabet $\Sigma = \{A, B, C, \ldots, Z\}$
- Elements of the analogy are words over Σ

Advantages of this micro-world

- Simplicity of the problems
- Human readibility
- Implies simple operations (predecessor, successor, add, remove, increment...)
- Covers a wide range of problems

Examples of Hofstadter's problems

- ABC : ABD :: IJK : x
- RST : RSU :: RRSSTT : x
- ABC : ABD :: BCA : x
- ABC : ABD :: AABABC : x
- IJK : IJL :: IJJKKK : x
- ...

Minimum Description Length Principle

One more time!

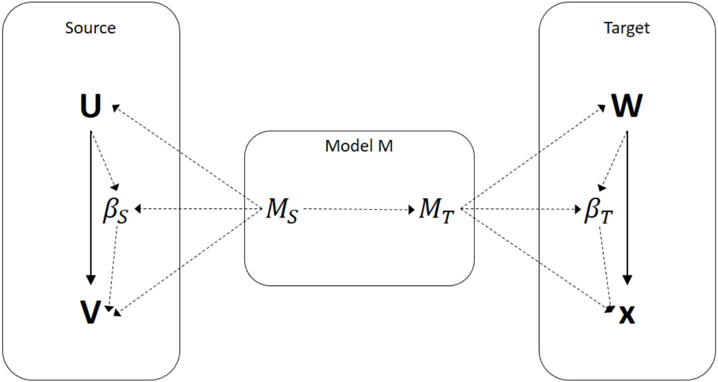
MDL Principle

The best theory to describe observed data is the one which minimizes the sum of the description length (in bits) of:

- the theory description
- the data encoded from the theory

Let's try to apply the MDL Principle to analogy reasoning!

A simplification
[Cornuéjols, 1998]



A new point of view

A descriptive language for analogies

```
// ABC : ABD :: IJK : IJL
let(alphabet, shift, ?, sequence, 3),
 let(mem,, ?, next_block, mem,, ?, last, increment),
 mem,,, next_block, mem,, 8;

// ABC : ABD :: IJK : IJD
let(alphabet, shift, ?, sequence, 3),
 let(mem,, ?, next_block, mem,, ?, last, ‘d’),
 mem,,, next_block, mem,, 8;
```

A new point of view

Results

Problem	Solution	Proportion	Complexity
IJK *16.0 ± 0.085 s*	IJL	93%	37
	IJD	2.9%	38
BCA *21.7 ± 0.12 s*	BCB	49%	42
	BDA	43%	46
AABABC *23.8 ± 0.12 s*	AABABD	74%	33
	AACABD	12%	46
IJKLM *24.7 ± 0.22 s*	IJKLN	62%	40
	IJLLM	15%	41
123 *6.39 ± 0.074 s*	124	96%	27
	123	3%	31
KJI *18.6 ± 0.13 s*	KJJ	37%	43
	LJI	32%	46
135 *9.93 ± 0.10 s*	136	63%	35
	137	8.9%	37
BCD *21.9 ± 0.30 s*	BCE	81%	35
	BDE	5.9%	44

Problem	Solution	Proportion	Complexity
IJJKKK *13.7 ± 0.11 s*	IJJLLL	40%	52
	IJJKKL	25%	53
XYZ *11.2 ± 0.093 s*	XYA	85%	40
	XYZ	4.4%	34
122333 *10.0 ± 0.098 s*	122444	40%	56
	122334	31%	49
RSSTTT *10.4 ± 0.072 s*	RSSUUU	41%	54
	RSSTTU	31%	55
IJJKKK *8.67 ± 0.071 s*	IJJLLL	41%	52
	IJJKKL	28%	53
AABABC *12.2 ± 0.12 s*	AABABD	72%	33
	AACABD	12%	46
MRRJJJ *22.1 ± 0.18 s*	MRRJJK	28%	64
	MRRKKK	19%	65
147 *13.6 ± 0.20 s*	148	69%	36
	1410	10%	38

Outline

Conclusion

What to remember?

- Complexity = compression
- Difference between deduction and induction
- Non-universality of inductive reasoning
- Toward a universal solution: Solomoff's lightsaber
- What is analogy reasoning?

References

Licence de droits d'usage

Contexte public } sans modifications