What is a pattern?
An introduction to a theory of grounded computation

Hiroshi Ishikawa
Nagoya City University

October 20, 2008. At INRIA Sophia Antipolis
Motivation
Example: Binary Restoration

Greig, Porteous, and Seheult ’89

\[
E(X) = \sum_{v \in V} \lambda |Y_v - X_v| + \sum_{(u,v) \in E} |X_u - X_v|
\]

- This is the optimal solution, given the smoothing prior.
- But it does not look perfect.
Higher-order Relationships

- Boundary smoothness
- Linearity
- Collinearity
- Parallelism
Higher-order Relationships

- Boundary smoothness
- Linearity
- Collinearity
- Parallelism
- Larger structure
- Regular structure
- Higher-level knowledge
Can we state the problem?

- We see “patterns” in these.
- How can we state the problem precisely?
- What space do these things live in?
What is a pattern?

The Problem
What is a pattern?

- Define patterns in these
From dictionaries:

**pattern**

- a combination of qualities, acts, tendencies, etc., forming a **consistent or characteristic arrangement**: “the behavior patterns of teenagers.”

- a model or original used as an **archetype**.

- a perceptual **structure**; “a visual pattern must include not only objects but the spaces between them”

[There are many more (less abstract) meanings. Only pertinent ones are shown here.]
In the case of strings

These strings “have patterns”

01010101010101010101010101010101010101010101010101010101...
0101101110111101111101111110111111...
1100100100001111110110101010001000...

• We can say a string with much less information than its length have pattern

• That is, if there is a much shorter program to print out the string
Two descriptions

• To describe a string by another string, there are two ways:
  – Describe the string by itself (*dense*)
  – By a program that prints out the string (*sparse*)
• The latter can exploit any *pattern* in the string
• If the string has a pattern, the latter description can be much shorter.
• The length of the shortest such description is called its Kolmogorov Complexity
• *Pattern* is computation
What about objects other than strings?

- Can we define a measure of information in these?
The Thesis of Computer Science

“Information can be encoded”

- Computation is only defined in the world of symbols and strings
- Everything else can be encoded into strings
  - Ex. Real numbers $\rightarrow$ binary expansion

\[
\begin{array}{c}
X \xrightarrow{f} B \\
\text{encode}
\end{array}
\]

Objects $\xrightarrow{\text{compose}}$ Strings

computation
Encoding the Universe

To define an information measure in general objects:

• Fix an encoding (an enumeration) of the objects
• Define the information in an object $x$ by the Kolmogorov Complexity $K(f(x))$ of its code string

$$
\begin{align*}
X & \xrightarrow{\text{encode}} B \xrightarrow{K} \mathbb{N} \\
\cup & \quad \cup \quad \cup \\
x & \quad f(x) & \quad K(f(x))
\end{align*}
$$
Example: Shapes on Euclidean Plane

Encode:

- Real number: binary expansion
- Point: a pair of reals
- Line: two points
- Circle: a point and a real
- …
Example: Shapes on Euclidean Plane

But then:

- How far must we go?
  - We can never exhaust all possible shapes
  - This is only one example

- We are presupposing what the patterns are

- The encoding $f$ is doing the actual computation

\[ X \xrightarrow{\text{encode}} f \xrightarrow{B} K \xrightarrow{\mathbb{N}} \]
Example: Shapes on Euclidean Plane

So let’s encode only the points

- Point $\rightarrow$ 2 reals $\rightarrow$ alternate binary expansion
  $$(0.x_0x_1x_2x_3..., 0.y_0y_1y_2y_3...) \rightarrow x_0y_0x_1y_1x_2y_2x_3y_3....$$
- Finite number of points $\rightarrow$ rotate among points
  $$a \rightarrow a_0a_1a_2a_3..., \quad b \rightarrow b_0b_1b_2b_3..., \quad c \rightarrow c_0c_1c_2c_3..., \quad \{a,b,c\} \rightarrow a_0b_0c_0a_1b_1c_1a_2b_2c_2a_3b_3c_3...$$
- Countable number of points $\rightarrow$ rotate among increasingly many points
  $$\{a,b,c,d,...\} \rightarrow a_0a_1b_0a_2b_1c_0a_3b_2c_1d_0...$$
- We can only encode a countable subset
So let’s only consider countable sets.

1. Require the coordinates of the points to be rational
   - Then any set of such points are countable
   - But many sets don’t have many rational points
     example: the line $y = \sqrt{2}x$

2. Only consider any countable sets
   - Even a single point can have an infinite information

Example: Shapes on Euclidean Plane
Fundamental Problems

• Hard to define general enough encoding $f$

• We need to be able to define the information amount in specific parts \textit{a priori}
  
  – Ex: Points to have the same amount of information

$$X \xrightarrow{f} B \xrightarrow{K} \mathbb{N}$$

\begin{align*}
X & \xrightarrow{f} B & K & \mathbb{N} \\
\cup & \xrightarrow{\text{encode}} & \cup & \xrightarrow{\text{ }} \cup \\
x & \xrightarrow{f(x)} & K(f(x)) & \\
\text{encode}
\end{align*}
The Arbitrariness Trap

• In CS, the encoding $f$ is just said to “exist”.
• The arbitrariness of $f$ makes the information measure arbitrary, too.
• So Computer Science does have to think about encoding seriously.
A Solution
Remember the two descriptions

• To describe a string by another string, there are two ways:
  – Describe the string by itself (dense)
  – By a program that prints out the string (sparse)

• Similarly, we will represent an object in two ways:
  – Represent it by a subset (dense representation)
    *Ground representation*
  – By a new representation (sparse representation) that produces the dense representation
    *Representation by diagrams and cross sections*
Ground Representation

- Something like bitmap, representing raw data
- Abstract a bit → subsets
  - Subsets of Euclidean plane
  - Image: a subset of $\mathbb{R}^2 \times Color$
    
    i.e., image function $I(p)$ on $D \subset \mathbb{R}^2$
    
    $\Leftrightarrow \{ (p, I(p)) \mid p \in D \} \subset \mathbb{R}^2 \times Color$
  - Objects in 3D space (at a level of abstraction) :
    
    Car $\subset \mathbb{R}^3 \times Material$
    
    $Material = \{ \text{iron, rubber, glass, plastic, \ldots} \}$
Ground Representation

• Raw signal
• *A priori*
• Almost everything can be easily represented
• Does not reflect the structure in the object, even if there is any
• Most that can be represented is random noise
Representation

• Specify sets and *power maps* between them (*diagram*)

  \[
  \begin{align*}
  \varphi &: \mathcal{P}(Y) \to \mathcal{P}(X) \\
  \psi &: \mathcal{P}(Z) \to \mathcal{P}(Y) \\
  \eta &: \mathcal{P}(W) \to \mathcal{P}(Y) \\
  \kappa &: \mathcal{P}(X) \to \mathcal{P}(W)
  \end{align*}
  \]

Ex.

• Consider an assignment \( s \) to each set of its subset that satisfies

\[
s(X) = \varphi(s(Y)) \\
s(Y) = \psi(s(Z)) \cap \eta(s(W)) \\
s(W) = \kappa(s(X))
\]

• Specify some of the \( s(.) \)

• Designate one set (say \( X \)) whose assigned subset \( (s(X)) \) is the dense representation of what is represented.
The Simplest Examples

\[ s(\mathbb{R}) = \{r\} \]
\[ s(X') = \{p\} \quad s(X) = ? \]

\[ s(S') = \bigcap_{\mu \in \text{in}(S)} \mu(s(\text{dm}(\mu))) \]

\[ f : A \rightarrow B \]
\[ \rightarrow \]
\[ f : \mathcal{P}(A) \rightarrow \mathcal{P}(B) \quad A \supset S \mapsto \{f(x) \mid x \in S\} \subset B \]
\[ f^{-1} : \mathcal{P}(B) \rightarrow \mathcal{P}(A) \quad B \supset S \mapsto \{x \in A \mid f(x) \in S\} \subset A \]

\( X, X' : \) Euclidean plane

\( \text{dist} : X \times X \rightarrow \mathbb{R} \) distance function

\( \pi_1, \pi_2 : \) Projection

\( \mathbb{R} \xrightarrow{\text{dist}^{-1}} X \times X' \)

\[ s(\mathbb{R}) = \{r\} \]

\[ s(X') = \{p\} \]

\[ s(X) = ? \]
The Simplest Examples

\[ \mathbb{R} \xrightarrow{\text{dist}^{-1}} X \times X' \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = ? \]

\[ s(S') = \bigcap_{\mu \in \text{in}(S)} \mu(s(\text{dm}(\mu))) \]

\[ s(X \times X') = \text{dist}^{-1}(s(\mathbb{R})) \cap \pi_2^{-1}(s(X')) \]

\[ = \{(x, y) \mid \text{dist}(x, y) \in s(\mathbb{R}), \pi_2(x, y) = p\} \]

\[ = \{(x, p) \mid \text{dist}(x, p) = r\} \]

\[ s(X) = \pi_1(s(X \times X')) \]

\[ = \{x \mid \text{dist}(x, p) = r\} \]

\(X, X':\) Euclidean plane

\(\text{dist}: X \times X \to \mathbb{R}\) distance function

\(\pi_1, \pi_2:\) Projection
The Simplest Examples

\[ X, X' : \text{Euclidean plane} \]
\[ V, V' : 2\text{D vector space} \]

\[ \text{mult: } V \times \mathbb{R} \rightarrow V \quad (v, c) \mapsto cv \]
\[ \text{sub: } X \times X \rightarrow V \quad (x, y) \mapsto x - y \]

\[ \pi_1, \pi_2 : \text{Projection} \]

\[ s(V) = \{v\} \]
\[ s(X') = \{p\} \]
\[ s(X) = ? \]

\[ s(V \times \mathbb{R}) = \{(v, c) \mid c \in \mathbb{R}\} \]
\[ s(V') = \{cv \mid c \in \mathbb{R}\} \]
\[ s(X \times X') = \{(x, p) \mid x - p \in s(V')\} \]
\[ s(X) = \{x \mid x = p + cv, c \in \mathbb{R}\} \]
A little extension

- Choose some of the sets

Ex.

- Assume the assignment $s$ satisfies for the chosen sets

$$s(S) = \bigcup_{\mu \in \text{in}(S)} \mu(s(dm(\mu)))$$

instead of

$$s(S) = \bigcap_{\mu \in \text{in}(S)} \mu(s(dm(\mu)))$$
Combination

\[ s(\mathbb{R}) = \{r\} \]
\[ s(\mathbb{R}) = \{r\} \]
\[ s(X') = \{p\} \]
\[ s(X') = \{p\} \]
\[ s(X) \]
\[ s(X) \]
\[ s(X') = \{q\} \]
\[ s(X') = \{q\} \]
\[ s(X) \]

\[ \pi_2^{-1} \]
\[ \pi_1 \]
\[ \text{dist}^{-1} \]

\[ \mathbb{R} \xrightarrow{\text{dist}^{-1}} X \times X' \]
\[ \pi_2^{-1} \]
\[ \pi_1 \]

\[ s(V) = \{v\} \]
\[ s(V) = \{v\} \]

\[ V \xrightarrow{\pi_1^{-1}} V \times \mathbb{R} \xrightarrow{\text{mult}} V' \]
\[ V \xrightarrow{\pi_1^{-1}} V \times \mathbb{R} \xrightarrow{\text{mult}} V' \]

\[ X' \xrightarrow{\pi_2^{-1}} X \times X' \xrightarrow{\pi_1} X \]
\[ X' \xrightarrow{\pi_2^{-1}} X \times X' \xrightarrow{\pi_1} X \]

\[ s(X') = \{q\} \]
\[ s(X') = \{q\} \]

\[ s(X) \]
\[ s(X) \]

\[ s(\mathbb{R}) = \{r\} \]
\[ s(\mathbb{R}) = \{r\} \]
\[ s(X') = \{p\} \]
\[ s(X') = \{p\} \]
\[ s(X) \]
\[ s(X) \]

\[ s(X') = \{q\} \]
\[ s(X') = \{q\} \]

\[ s(X) \]
\[ s(X) \]
Combination

\[ \mathbb{R} \xrightarrow{\text{dist}^{-1}} X \times X' \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(V) = \{ v \} \]

\[ s(X') = \{ q \} \]

\[ s(X) \subset X \]

\[ \mathbb{R} \times X' \]

\[ V \xrightarrow{\pi_1^{-1}} V \times \mathbb{R} \xrightarrow{\text{mult}} V' \]

\[ X' \xrightarrow{\pi_2^{-1}} X \times X' \]

\[ s(X') = \{ q \} \]

\[ s(X) \subset X \]

\[ \mathbb{R} \times X' \]

\[ p \]}

\[ q \]

\[ r \]
Recursive definition

\[ s(V) = \{ v \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = ? \]

\[ s(X \times V) = \{(x, v) \mid x \in s(X) \} \]

\[ s(X) = s(X') \cup \text{add}(s(X \times V)) \]

\[ = \{ p \} \cup \{ x + v \mid x \in s(X) \} \]

\[ \supset \{ \cdots, p - 2v, p - v, p, p + v, p + 2v, \cdots \} \]

\[ X, X' : \text{Euclidean plane} \]

\[ V : 2D \text{ vector space} \]

\[ \text{add} : X \times V \rightarrow X \quad (x, v) \mapsto x + v \]

\[ \pi_1, \pi_2 : \text{Projection} \]
Recursive definition

$s(V) = \{u,v\}$

$s(X') = \{p\}$

$s(X) = ?$

$V \xrightarrow{\pi_2^{-1}} X \times V$

$X' \xrightarrow{\text{id.}} X$

$X, X'$: Euclidean plane

$V$: 2D vector space

add: $X \times V \xrightarrow{\text{add}} X \quad (x, v) \mapsto x + v$

$\pi_1, \pi_2$: Projection
Hierarchical definition

\[ V \xrightarrow{\pi_2^{-1}} X \times V \]
\[ s(V) = \{u,v\} \]
\[ X' \xrightarrow{id.} X \]
\[ s(X') = A \]
\[ s(X) = ? \]

\[ X, X' : \text{Euclidean plane} \]
\[ V: 2D \text{ vector space} \]
\[ \text{add}: X \times V \to X \quad (x, v) \mapsto x + v \]
\[ \pi_1, \pi_2 : \text{Projection} \]

\[ A \subset X \]
Hierarchical definition

\[ s(\mathbb{R}) = \{r\} \]
\[ s(X') = \{p\} \]
Hierarchical definition

\[ \mathbb{R} \xrightarrow{\text{dist}^{-1}} X \times X' \]
\[ V \xrightarrow{\pi_2^{-1}} X \times V \]
\[ X' \xrightarrow{\pi_1^{-1}} \{u,v\} \]

\[ s(\mathbb{R}) = \{ r \} \]
\[ s(X') = \{ p \} \]
Computation

- Ex: Factorials

\[
\mathbb{N} \times \mathbb{N} \xrightarrow{} \mathbb{N} \times \mathbb{N} \cup (n, m) \mapsto (n + 1, m(n + 1))
\]

\[s(\mathbb{N} \times \mathbb{N}) = \{(0, 1)\}\]

\{(0, 1), (1, 1), (2, 2), (3, 6), \ldots, (n, n!), \ldots\}

- Ex: Fibonacci numbers

\[
\mathbb{N}^+ \times \mathbb{N}^+ \xrightarrow{id} \mathbb{N}^+ \times \mathbb{N}^+ \xrightarrow{\pi_1} \mathbb{N}^+ \cup (n, m) \mapsto (m, n + m)
\]

\[s(\mathbb{N}^+ \times \mathbb{N}^+) = \{(1, 1)\}\]

\\{1, 1, 2, 3, 5, 8, \ldots\}

- Can emulate any Turing machine
Computation

- Can embed computation in any space
Information Measure

• **Define** the complexity of any subset by the size of the **minimum diagram** that represents it.

• The size of a diagram is measured by the total size of maps **generated** from a fixed set of structure maps.

- $\text{id}: X \to X,$ $\pi_i: X_1 \times \ldots \times X_n \to X_i,$ $\omega: X \to \{0\}$
- $f: X \to Y,$ $g: Y \to Z \Rightarrow g \circ f: X \to Z$
- $f_i: X \to Y_i \Rightarrow f_1 \times \ldots \times f_n: X \to Y_1 \times \ldots \times Y_n$
Information Measure

• The information measure of a subset $A \subset X$ relative to the set of maps $\mathcal{M}$ is denoted by

$$I(A|\mathcal{M})$$

• But any subset can be represented by the trivial representation by defining the given set as the fixed partial cross section part of the representation.

$$s(X) = A$$

by defining the given set as the fixed partial cross section part of the representation.

• No map is needed, so the size is 0.
Information Measure

Consider constant $x \in X$ as a map $x : \{0\} \to X$, only allow the fixed partial cross section $s(\{0\}) = \{0\}$

$I(C \mid \{r,p,\text{dist}\}) \leq 5$

$I(L \mid \{v,q,\text{mult, sub}\}) \leq 7$
Information Measure

We choose the maps and constants that characterize the structure of the space in question. For instance, for a Euclidean space $X$, it might be

$$\mathcal{M}_E = \{\text{dist, add, sub, mult}\} \cup X \cup V \cup \mathbb{R}$$

Then,

- $I(C \mid \mathcal{M}_E) \leq 5$, $I(L \mid \mathcal{M}_E) \leq 7$
- For any finite set $A \subset X$, $I(A \mid \mathcal{M}_E) \leq |A|$
Relation with Kolmogorov Complexity

**Theorem** For any Turing machine $U$, there exists a diagram $D_U$ generated by $M_N$ including the sets $S = \mathbb{N} \times \mathbb{N}$ and $T = \mathbb{N} \times \mathbb{N}$ such that, for any $\sigma \in b^*$ and any cross section $s$ with $s(S) = \bar{\sigma}$, $s(T) = \emptyset$ if $U(\sigma) = \uparrow$ and $s(T) = \bar{\tau}$ if $U(\sigma) = \tau \in b^*$.

$s(S) = \bar{\sigma}$  \hspace{2cm} $s(T) = \bar{\tau}$

$\sigma = 0101$ \hspace{1cm} $U(\sigma) = \tau$ \hspace{1cm} $\tau = 01100$
Relation with Kolmogorov Complexity

For \( b = \{0, 1\} \), \( \sigma \in b^* \), define \( \overline{\sigma} \subset \mathbb{N} \times \mathbb{N} \) as

\[
\overline{\sigma} = \{(i, \sigma[i]) \mid i = 0, 1, \ldots, |\sigma| - 1\} \cup \{(|\sigma|, 0), (|\sigma|, 1)\}
\]

Ex: If \( \sigma = 0110 \), \( \overline{\sigma} = \{(0,0), (1,1), (2,1), (3,0), (4,0), (4,1)\}

\[
\begin{array}{ccccccc}
0 & 1 & 2 & 3 & 4 & 5 & \cdots \\
0 & & & & & & \\
1 & & & & & & \\
\vdots & & & & & & \\
\end{array}
\]

\( \mathcal{M}_\mathbb{N} = \{0, \text{succ}\} \)

- \( 0 : \{0\} \rightarrow \mathbb{N} \), \( 0(0) = 0 \)
- \( \text{succ} : \mathbb{N} \rightarrow \mathbb{N} \) \[\text{succ}(n) = n+1\]
Relation with Kolmogorov Complexity

Part of the diagram emulating a Turing machine
Relation with Kolmogorov Complexity

**Theorem** There exists a Turing machine that, given an encoded diagram $D_\sigma$ representing $\bar{\sigma}$ for $\sigma \in b^*$, outputs $\sigma$ and terminates.

Since $D_\sigma$ can be encoded with size $\sim I(\bar{\sigma} | \mathcal{M}_\mathbb{N})$, $K(\sigma)$ is dominated by a constant multiple of $I(\bar{\sigma} | \mathcal{M}_\mathbb{N})$.
Automatic handling of structures

- Implement basic components approximately: spaces and maps
  - real numbers, spaces, ...
- Structural varieties are generated combinatorially
- The structure does not depend on the approximation.
  - Data exchange between systems does not change the structure itself
Pattern Discovery as Optimization

• Given a signal-level data, find ever smaller diagrams representing it.
• The upper bound is the trivial representation
• Regular parts can be represented more compactly
Concluding Remarks

• Patterns and computations are defined directly
• Can mix sparse and dense representation
• Hierarchical and recursive representation possible
• Connects the representation to the represented. No arbitrariness in encoding across structures.
• The representation is a formalization of definition; i.e., a meta-definition.
• We can treat consistently and automatically what we usually define in natural languages