A theory of incremental compression

Arthur Franz
Independent researcher

Email: franz@fias.uni-frankfurt.de

Motivation

• AGI has to construct models / descriptions of the world given its observations: induction

• Main principle: Occam’s razor. Concise, parsimonious, short

• Solomonoff’s universal induction has solved the problem in theory.

• How to build a general AI that finds good descriptions of the world in a wide range of environments?

• Universal Search is general but inefficient: $2^{l(p)+1}t(p)$
  0, 1, 00, 01, 10, 11, 000, 001, 010, 011,...

• Current efficient approaches are not general (PCA, deep learning): “narrow AI”

For AGI:
How to make things practical while still staying general?
Describing stuff in practice is incremental

• Objects have features: size, shape, color, location,...

Object encoding, e.g. as binary string

- Size spec
  - Everything except size
- Shape spec
  - Everything except size, shape
- Color spec
  - Everything except size, shape, color
- Location spec

• Features are independent properties
• If features are described concisely, the full description will be concise as well: Shortest description = size + shape + color + location

• General mathematical theory of incremental compression is presented here
• Exponential speed up compared to Levin Search, while still general
A single compression step

\[ f'(x) = p \quad f(p) = x \]

\[ f \quad p \]

feature parameters

finite strings \leftrightarrow \text{integers} \leftrightarrow \text{Turing machines}

Goal: Find shortest feature \( f^* \) and descriptive map \( f'^* \) such that \( f^*(f'^*(x)) = x \) and some compression has been achieved: \( l(f^*) + l(p) < l(x) \), where \( p \equiv f'^*(x) \)
Full compression scheme

Example:

\[ x = 10110111011110111110 \ldots \]

\( f_1 \) prints \( m \) 1’s and attaches a 0

\[ p_1 = 1, 2, 3, 4, 5, \ldots \]

\( f_2 \) starts with \( p_2 = 1 \) and counts up
Features and parameters are independent

1. $f^*$ is incompressible: $K(f^*) = l(f^*) + O(1)$

2. Representation of $x$ breaks down into an independent feature and its parameters:

   $K(x) = K(f^*, p) = K(f^*) + K(p) = l(f^*) + K(p)$ up to additive constants.

Intuition:
If the feature and its parameters contained common information, the feature could be made even shorter. It doesn't need that information since it takes it along when computing $x$. 
**Orthogonal feature basis**

The shortest features \( \{f_i^*\} \) constitute a complete basis of \( x \):

\[
K(x) = l(f_1^*) + K(p_1) = l(f_1^*) + l(f_2^*) + K(p_2) = \cdots = \sum_{i=1}^{k} l(f_i^*)
\]

Since \( p_i \) is independent of \( f_i^* \), all the following features are as well. Thus, the mutual algorithmic information between features is zero:

\[
I(f_i^*|f_j^*) = 0 \text{ for all } i \neq j
\]

\( \{f_i^*\} \) is an orthogonal basis of \( x \)

Efficiency: \( 2^{l(f^*)} \approx 2^{K(x)/k} \ll 2^{K(x)} \)

What about \( l(f'^{*}) \)?
Bound on the length of the descriptive map

\[ l(f'^*) \leq \log K(x) + 2 \log \log K(x) + O(1) \]

\[ 2^{l(f'^*)} \lesssim K(x) \ll 2^K(x) \]

In general:

\[ K(p) \leq K(p|z) + K(z) + O(1) \]

Condition on \( x \) and set \( z = K(x) \):

\[ K(p|x) \leq K(p|K(x), x) + K(K(x)|x) + O(1) \]

\[ \downarrow \quad \downarrow \]

\[ l(f'^*) \quad O(1) \]

Since

\[ K(K(x)|x) \leq \log K(x) + 2 \log \log K(x) + O(1) \]

the claim follows.
Efficiency of incremental compression

Time complexity, if universal search is used to find the feature and descriptive map:

\[ O \left( 2^{l(f^*)} + l(f^*) \right) \leq O \left( K(x) (\log K(x))^2 2^{l(f^*)} \right) \]

Time complexity of a potential algorithm for incremental compression goes like:

\[ \sim \sum_{i=1}^{k} 2^{l(f^*_i)} \]

Recall that \( K(x) = \sum_{i}^{k} l(f^*_i) + O(1) \)

Therefore, for non-incremental search we get, up to multiplicative constants:

\[ 2^{K(x)} = \prod_{i=1}^{k} 2^{l(f^*_i)} \gg \sum_{i=1}^{k} 2^{l(f^*_i)} \]

Incremental search is much more efficient than non-incremental search!
Discussion and Outlook

Is this practical?
• Already demonstrated: General incremental compression in my last year’s AGI paper

Realistic applications?
• Better AIXI approximation? Beyond Tic Tac Toe and Pac Man
• 100-bit worlds

Future work:
• How to find features?
• Hierarchical compression

Incremental compression is fast, general and much closer to practical application than (non-incremental) Universal Search
Thank you!