An Information Perspective on Evolutionary Computation

Yossi Borenstein
University of Essex

GECCO'08, July 12-16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-131-6/08/07

Two Measures of Information

Overview

- Two Measures of Information
  - Entropy
  - Kolmogorov Complexity
- Optimization & Information (Part I)
  - Entropy implies bounds!
    - NFLTs as a specific case
  - Kolmogorov Complexity implies strong statistical properties!
- Optimization & Information (Part II)
  - Entropy implies bounds!
    - NIAH as a specific case
  - Kolmogorov Complexity implies bad performance!
- Conclusion

Entropy

- Defined for: *Probability distributions*

Let $X = \{x_1, x_2, \dots, x_n\}$,
Let R be a random variable taking values in X
with distribution , $P(R = x) = p_x$

$$H(P) = \sum_{x \in X} p_x \log 1/p_x$$

Entropy

- Defined for: *Probability distributions*
- Represents the **uncertainty** about the outcome:
  - Degenerate Distribution:

$$P(x) = \begin{cases} 1 & if\ x = x' \\ 0 & otherwise \end{cases} \quad \Rightarrow \quad H(P) = 0$$

  - Uniform Distribution:

$$P(x) = 1/n \quad \Rightarrow \quad H(P) = \log n$$

Kolmogorov Complexity

- Defined for: *a single object*

Let $x = \{0,1\}^n$ be a binary string of length n,

The KC is a function : $K : \{0,1\}^* \Rightarrow N$

It representes the *size* of the minimal binary representation of a program that can generate x

Optimization Scenario

- Problem as a distribution over Instances
- Entropy as a measure of expectation.

Let $F = \{f_1, f_2, ..., f_N\}$,
Let R be a random variab ≥ le taking values in F
with distribution , $P(R = f) = P_f$

$$0 \leq H(P) \leq \log N$$

Kolmogorov Complexity

- Defined for: *a single object*
- Represents: the “regularity” of an object
  - *S=“$\overbrace{000000000\ldots0}^{n}$”*

begin
for $(i = 0 \text{ to } n)$ print `0`; $\quad \Rightarrow \quad K(S) = O(\log n)$
end

Kolmogorov Complexity

- Defined for: *a single object*
- Represents: the "regularity" of an object
  - $S = \text{"}\overbrace{011010110\ldots0}^{n}\text{"}$

```
begin
  print `011011010...`;
end
```

$\Rightarrow \quad K(S) = O(n)$

Almost all strings are Incompressible

There are 2^n possible binary strings

But only $\sum_{i=0}^{n-1} 2^i = 2^n - 1$ shorter descriptions

- At least one string cannot be compressed at all!
- More generally:
  - *At least one-half(!) are 1-incompressible*
  - *At least three-fourth are 2-incompressible*
  - *At least* $1 - 2^{-k}$ *are k-incompressible*

- Incompressibility imposes strong statistical properties

Almost all strings are Incompressible

The are 2^n possible binary strings of length n

But only:

- 2 "programs" of length 1 {"0", "1"}
- 4 "programs" of length 2 {"00", "01", "10", "11"}
- $\vdots$
- 2^{n-1} "programs" of length n-1 {...}

$$= \sum_{i=0}^{n-1} 2^i = 2^n - 1 \text{ shorter descriptions}$$

Kolmogorov Complexity of Functions

$$K(f) = \min_{p_f \in \{0,1\}^*} \{ l(p_f) : \forall x \in X, p_f(x) = f(x) \}$$

$$Const(x) = a \quad \Rightarrow \quad K(Const) = O(1)$$

$$NIAH(x) = \begin{cases} 1 & if \ x = x_{opt} \\ 0 & otherwise \end{cases} \quad \Rightarrow \quad K(NIAH) = O(n)$$

$$Rand(x) = \begin{cases} 123 & if \ x = x_0 \\ 24 & if \ x = x_1 \\ \vdots & \vdots \end{cases} \quad \Rightarrow \quad K(Rand) = O(2^n \log 2^n)$$

Optimization Scenario

- No a priori knowledge!
- Learning a function "on the go"

$$\{\{x_{a1}, f(x_{a1})\}, \{x_{a2}, f(x_{a2})\}, \{x_{a2}, f(x_{a2})\}\} \Rightarrow ?$$

- KC measure how easy it is to extrapolate.

Preliminaries

Let $f : X \to Y$ where X, Y are finite sets.
Let F denote all such functions.

A non - repetetive deterministic search algorithm a is represented by a trace :

$$a(f): \begin{matrix} x_{a1} \\ f(x_{a1}) \end{matrix}, \begin{matrix} x_{a2} \\ f(x_{a2}) \end{matrix}, \begin{matrix} \cdots \\ \cdots \end{matrix}, \begin{matrix} x_{an} \\ f(x_{an}) \end{matrix}$$

Performance is a function of the trace!

Optimization & Information

PART I

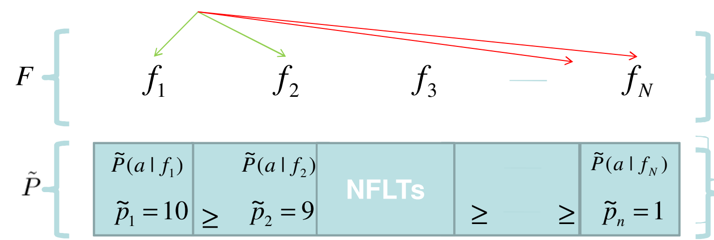
Entropy

NFLTs = *max* Entropy

- *"All search algorithms are equivalent when compared over all possible discrete functions."* Wolpert, Macready (1995)

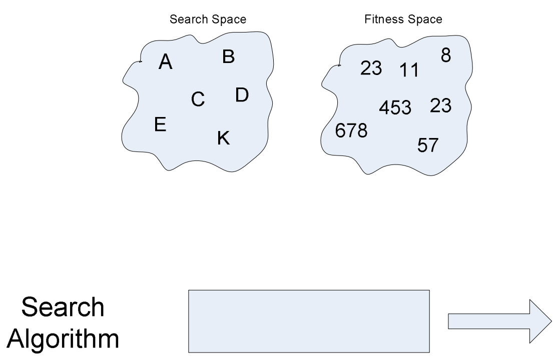
$$\text{Let } f: X \to Y, \quad Y = \{y_1, y_2, \ldots, y_n\},$$
$$\text{For all } x \in X, y \in Y:$$
$$\Pr(f(x) = y \mid f(x_{a1}), \ldots, f(x_{ak})) = 1/(n-k)$$

Entropy Implies Bounds



Entropy	**Best**	$\geq E[\tilde{P}(a \mid H(P) = h)] \geq$	**Worst**
$H(P) = 0$	10		1
$H(P) = 1$	$(10+9)/2 = 9.5$		$1.5 = (1+2)/2$
$H(P) = \log N$	$avg(10 \ldots 1) = 5.5$		$5.5 = avg(10 \ldots 1)$

(Sharpen) No Free Lunch



Entropy Implies *good* Bounds?

$$\text{let } x \in X \text{ and } \pi: X \to X \text{ a random permuation:}$$

$$f_{1\max}(x) = \sum \delta(x_i = 1) \qquad \left\{ f_{\max}^{x'}(x) = \delta(x'_i = x_i) \right\}$$

$$f_{1rand}(x) = f_{1\max}(\pi(x)) \qquad \left\{ f_{rand}^{x'}(x) = f_{\max}^{x'}(\pi(x)) \right\}$$

Conjecture:

$$\left\{ f_{\max}^{x'}(x) \right\} \text{ can be solved more efficiently than } \left\{ f_{rand}^{x'}(x) \right\}$$

Entropy Implies *good* Bounds?

$$P_{rand}(f) = \begin{cases} 1/2^n & \text{if } f \in \{f_{rand}^{x'}(x)\} \\ 0 & \text{otherwise} \end{cases}$$

$$P_{\max}(f) = \begin{cases} 1/2^n & \text{if } f \in \{f_{\max}^{x'}(x)\} \\ 0 & \text{otherwise} \end{cases}$$

$$\Rightarrow \quad H(P_{\max}) = H(P_{rand})$$

Conjecture:

$\{f_{\max}^{x'}(x)\}$ can be solved more efficiently than $\{f_{rand}^{x'}(x)\}$

Kolmogorov Complexity

- While the entropy of the two classes is the same, the KC is clearly different!

$$K\left(\left\{f_{\max}^{x'}(x) = \sum \delta(x_i = x'_i)\right\}\right) \approx O(\log n)$$

$$K\left(\left\{f_{rand}^{x'}(x) = f_{\max}^{x'}(\pi(x))\right\}\right) \approx O(\log(2^n!))$$

Kolmogorov Complexity

Conservation of Information

- The algorithm cannot contribute more information than it contains.

Let $f = \{f(x_1), \quad f(x_2), \quad \cdots, \quad f(x_n)\}$

Let $a_f = \{f(x_{a1}), \quad f(x_{a2}), \quad \cdots, \quad f(x_{an})\}$

$$|K(f) - K(a_f)| \leq K(a)$$

- The KC of a search algorithm is (almost) constant!

Conservation of Information

Let $a_{f\,\max} = \left\{ f(x_{a_{f\,\max}1}),\quad f(x_{a_{f\,\max}2}),\quad \cdots,\quad f(x_{a_{f\,\max}n}) \right\}$

Let $a_{f\,\text{rand}} = \left\{ f(x_{a_{f\text{rand}}1}),\quad f(x_{a_{f\text{rand}}2}),\quad \cdots,\quad f(x_{a_{f\text{rand}}n}) \right\}$

Incompressible!

$$K(a_{f\,\max}) \approx K(a)$$

$$K(a_{f\,\text{rand}}) >> K(a)$$

Difficult to Interpret!

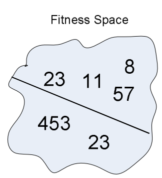
$a_{f\,\text{rand}} = \left\{ f(x_{a_{f\text{rand}}1}),\quad f(x_{a_{f\text{rand}}2}),\quad \cdots,\quad f(x_{a_{f\text{rand}}n}) \right\}$

The Incompressibility Method

$a_{f\,\text{rand}} = \left\{ f(x_{a_{f\text{rand}}1}),\quad f(x_{a_{f\text{rand}}2}),\quad \cdots,\quad f(x_{a_{f\text{rand}}n}) \right\}$

$\log(|Y|) = \log(2^n) = n$ | n | $\cdots$ | n

Can a short search algorithm sample with high frequency solutions above median?

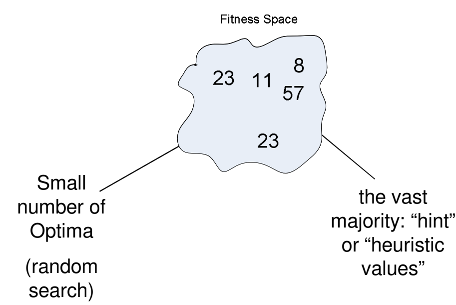


No!
This will imply a way to compress the trace!

$\log(2^n / 2) = n - 1$ | $n-1$ | $\cdots$ | $n-1$

Preliminaries (II)

- Objective: sample particular (optimal) solutions



Optimization & Information

PART II

(From *fitness* information to *spatial* information)

Example

- (1+1) EA (or any other search algorithm) using *tournament selection*

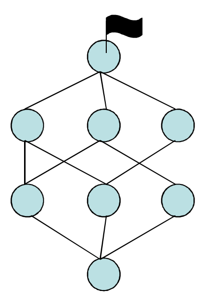
1. Choose x uniformly from $\{0,1\}^n$
2. Repeat :
   2.1 $x' := x$. Flip each bit of x' with probability $1/n$.
   2.2 $\mathit{if}\ \mathit{tournament}(x', x) = x'$ then $x := x'$.

From *fitness* information to *spatial* information

The search algorithm uses "*heuristic values*" to infer the **position** of optima

Fitness Spatial

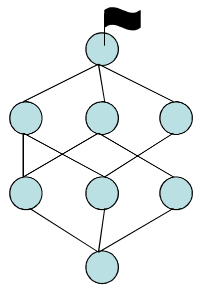
- Step I: define a graph (e.g., Hamming distance)



Fitness → Spatial

- Step II: the algorithm uses the "heuristic values" to define the probability distribution

$$t(x,y) \equiv \Pr(x \mid x,y) = \begin{cases} 1 & \text{if } f(x) > f(y) \\ 0.5 & \text{if } f(x) = f(y) \\ 0 & \text{if } f(y) > f(x) \end{cases}$$



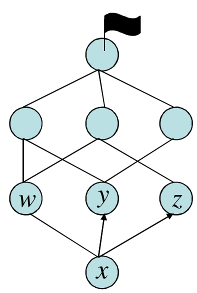
Entropy, KC & Performance

Fitness → Spatial

- Step III: the algorithm uses this probability to define the rule, e.g.,

$$(x,w) \Rightarrow x$$
$$(x,y) \Rightarrow y$$
$$(x,z) \Rightarrow z$$
$$\vdots$$

- Step IV: refine the graph using the rule
- Step V: Walk!



Fitness → Spatial

Fitness Function ⇨ **Step II:** **Information Content** ⇨ **Step III:** **Possible Decision Matrix**

X	f(x)
000	6
001	5
010	5
011	3
100	2
101	2
110	2
111	7

X	000	001	010	011	100	101	110	111
000		1	1	1	1	1	1	0
001			0.5	1	1	1	1	0
010				1	1	1	[illegible]	0
011					1	1	1	0
100						0.5	0.5	0
101							0.5	0
110								0
111								

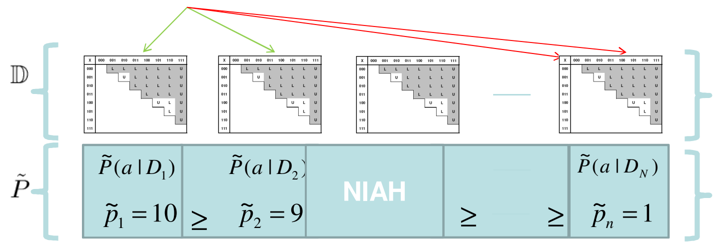
$$H(P_f) = \sum \delta(t(x,y) = 0.5)$$

X	000	001	010	011	100	101	110	111
000		L	L	L	L	L	L	U
001			U	L	L	L	L	U
010				L	L	L	L	U
011					L	L	L	U
100						U	L	U
101							L	U
110								U
111								

D

$$P_f(D) \equiv \Pr(D \mid f) = \prod_{x,y \in X} \big(D(x,y) = t(f(x), f(y))\big)$$

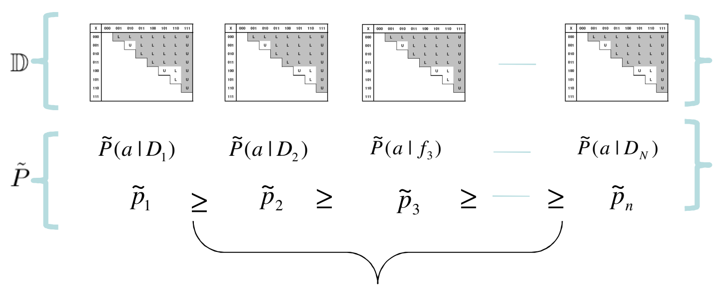
Entropy Implies Bounds even for a single function!



Entropy	**Best**	$\geq E\left[\tilde{P}(a \mid H(P_f)=h)\right] \geq$	**Worst**
$H(P_f)=0$	10		1
$H(P_f)=1$	$(10+9)/2=9.5$		$1.5=(1+2)/2$
$H(P_f)=\log N$	$avg(10...1)=5.5$		$5.5=avg(10...1)$

Almost all *D*'s are Incompressible!

- Simply consider the equivalent binary representation



Almost all the possible performance values are equal!

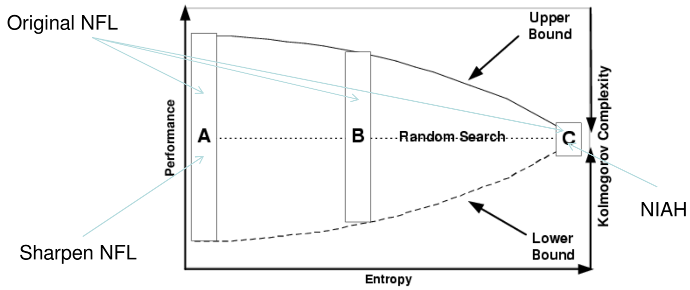
Conclusions

Entropy

- Distribution of instances (or functions):
  - Implies bounds
  - NFLTs are a special case
- Distribution of decision matrices
  - Implies bounds
  - NIAH is a special case
- Each permutation closure of a single function is associated with a value of entropy

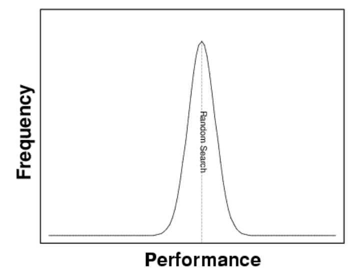
Entropy

- The higher the entropy of P_f the closer the performance to that of a random search.



Kolmogorov Complexity

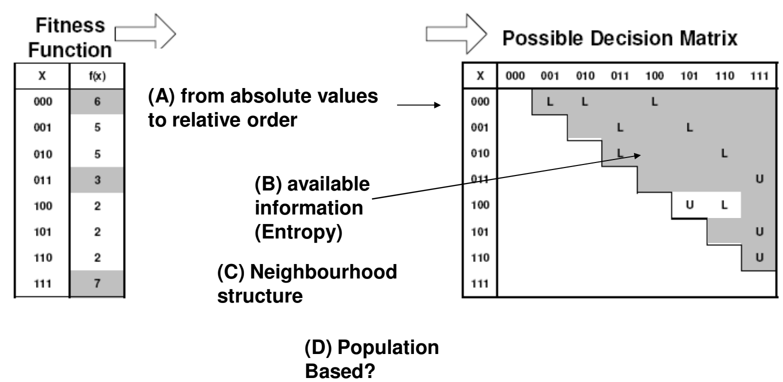
- *Almost all the performance values are identical and equals* the expected performance of a random search.*



Kolmogorov Complexity

- Implies strong statistical properties on the sampled fitness values
- High Kolmogorov complexity implies hardness
- The vast majority of all decision vectors are incompressible (and therefore) hard!
- Some fitness functions are intrinsically hard!

Some Thoughts… Information Reduction



Concluding remarks

- This tutorial focused only on two aspects of difficulty (KC and entropy)
  - Naturally, more criteria exist.
- The relation to KC and hardness is not straight forward. Any interpretation based on the KC of a fitness function should be very cautious.

References/Further reading

- D.H. Wolpert and W.G. Macready. *No free lunch theorems for optimization*. IEEE Trans Evolutionary Computation, 4:67–82, 1997.
- C. Schumacher and M. Vose and D. Whitley. *The No Free Lunch and Problem Description Length*. GECCO 2001.
- W.G. Macready and D.H. Wolpert. *What makes an optimization problem hard?*. Complex , 1:5:40--46, 1996.
- P. Grünwald and P. Vitanyi. *Algorithmic Complexity*. Handbook on the Philosophy of Information. To appear.
- T. English. *Optimization is Easy and Learning is Hard In the Typical Function*. Proc. 2000 Congress on Evolutionary Computation (CEC 2000) , pages 924--931, 2000.
- H. Buhrman, M. Li, J. Tromp, P. Vitanyi . *Kolmogorov Random Graphs And The Incompressibility Method*. SIAM Journal on Computing, 29:590—599.
- Y. Borenstein, R. Poli. *Kolmogorov complexity, Optimization and Hardness*, IEEE CEC 2006
- Y. Borenstein, R. Poli. *Information Perspective of Optimization*. PPSN 2006: 102-111.
- Y. Borenstein. *What Makes an Optimization Problem Hard? An Information-Theoretic Perspective*. PhD Thesis, Chapter 3.