Komolgorov Complexity

CS154
Chris Pollett
May 10, 2006.

Outline

- Minimal Description Length
- Inequalities
- Optimality
- Incompressible Strings

Information

- Today we would like to define a notion of how much information a string contains.
- This is also related to how random a string is.
- This area is useful in the development of pseudo-random number generators which can be used in randomized algorithms and in cryptography.

Descriptions via TMs

- Consider the following string:
  11111111111111111111111111111111111111111111111
  11111111111111111111111111111111111111111111111
  11111111111 11111111111111111111
- Although it is quite long it is not very random.
- I can quite simply describe it as write hundred and twenty five 1's in a row.
- One can make a TM *M* which reads its input in binary then writes out that many 1's.
- If I had a million 1's. Then <*M*, 1000000> would easily be shorter to write down then all these 1's.
- This motivates the next definition.

Minimal Description Length

Definition. Let x be a binary string. The minimal description of x, $d(x)$, is the lexicographical first shortest string $<M, w>$ such that M on input w halts with x on the tape. We define $K(x) = |d(x)|$.

Theorem. $\exists c \forall x[K(x) \leq |x| + c]$.

Proof. Let M be the machine that halts as soon as it starts. Then $<M, x>$ describes the string x. The length of $|M|$ is some number c. From which we get that $K(x) \leq |<M, x>| \leq |x| + c$ as desired.

Some Inequalities Involving Minimal Description Length

Theorem. $\exists c \forall x [K(xx) \leq K(x) + c]$.

Proof. Let $d(x) = <M, w>$ be a minimal description of x. Then $<N, <M,w>>$ describes xx, where N is the machine which on input $<M, w>$ runs M on w, then write the output of the simulation twice.

- Using the same kind of idea one can show:

Theorem. $\exists c \forall x,y\ [K(xy) \leq 2K(x) + K(\mathrm{y}) + c]$.

Optimality of the Definition

- Our definition is in terms of TMs, would it have made a difference to define minimal description length in terms of C++ programs?
- Let $K_p(x)$ be defined the same as $K(x)$ but using the description language p.

Theorem. $\forall x[K(x) \leq K_p(x) + c]$.

Proof. Consider the machine M which on input w simulates the programming language p on input w, then outputs what that programming language would output. So $<M, d_p(x)>$ outputs x and this string is at most constantly longer than $K_p(x)$.

Incompressible Strings

Definition. Let x be a string. Say that x is ***c*-compressible** if

$$K(x) \leq |x| - c.$$

If x is not c-compressible, we say that it is **incompressible by c**.

If x is not 1-compressible, we say that it is **incompressible**.

Theorem. Incompressible strings of every length exist.

Proof. The number of strings of length n is 2^n. Each description is a binary string, so the number of descriptions of length less than n is at most the sum of the number of strings of each length up to n-1, or

$$1 + 2 + 4 + 8 + + 2^{n-1} = 2^n - 1$$

which is less than the number of strings of length n. So some incompressible string of length n must exist.