Information Theory

Problem Set 08 - Kolmogorov Complexity and Universal Probability

Luís Felipe Ramos Ferreira

lframos_ferreira@outlook.com

1. (a) The Kolgomorov complexity of a string s is the length of the shortest program that can be passed to a UTM (Universal Turing Machine), so that the UTM outputs the string s and then halts.

   (b) A string is considered truly random when it's Kolgomorov complexity is bigger tah or equal to it's own length. A string that is truly random, for example, is the result string of subsequent flip of a coin. Since the flip of the coin is random, there is no way we can compress it using some kind of algorithm. A string that looks random but is not is the decimal expansion of the irrational number Pi.

   (c) The universal probability of a string s is the probability that, when we give a random program to a UTM, s is the output of the execution of such program. It is related to its Kolmogorov complexity by the equation $P_U(s) \approx 2^{-K(s)}$, where $P_U(s)$ is the universal probabilty of the string s and $K(s)$ is the Kolgomorov complexity of the string s.

2. We can construct a program that outputs the concatenation xy in the following way. First, use the fact that the program that describes x, which has at most $K(x)$ bits and the program that describes y has at most y bits. Then, create a program that first outputs x, using at most $K(x)$ bits, and then outputs y, which uses at most $K(y)$ bits. The piece of code that tells the order of the output is constant size and do not depend on the strings. So, the created program, which outputs the concatenation xy, has a Kolgomorov complexity of at most $K(x) + K(y) + c$.

3. (a) We can construct a program that first describes the number n_1 using at most $K(n_1)$ bits, then uses the language of the program that represents the sum operator (for example, the operator '+'), and then uses the string that represents the number n_2 using at most $K(n_2)$ bits. Since the sum operator described have a constant size, the description we showed has at most $K(n_1) + K(n_2) + c$ bits and describes number that is the sum of numbers n_1 and n_2.

   (b) As seen in the classes, strings generated by the flip of a fair coin are complex, since we can't describe them in a algorithmic way. In this scenario, we can flip a fair coint n times and create two strings based on the values of the flip. If the flip is heads, we put a 1 in the string n_1 and a 0 in the string n_2. Otherwise, we do the opposite. Both strings are complex, since they were created by the random flip of a fair coin. But if we add them, considering the binary number they represent, we achieve, by construction, a string of $N1's$, which is very simple and can be described very easily.

4. (a) a
   (b) b

References

[1] David J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. 7th edition, 2005.