Complexity Preserving Functions

Jan LEMEIRE
Vrije Universiteit Brussel (VUB)
Brussels, Belgium
Email: jan.lemeire@vub.ac.be

Extended Abstract for Workshop Complexity and Inference, DIMACS Center, New Jersey, June 2-5, 2003

Abstract
It's widely recognized that compression is useful and even necessary for inductive learning, where a short description will capture the 'regularities'. We introduce *complexity-preserving* functions that preserve these regularities of the concept. They are based on the universal information distance [Bennett et al. '97] and define for an instance a set of elements sharing the same complexity type. This corresponds to the two-part code [Rissanen '89] of the MDL principle, when it is interpreted as the first term describing the set and the second term the element in the set [Rissanen 99]. We investigate its importance in inductive learning.

Introduction
The Kolmogorov complexity of a string x $K(x)$ is the length of a shortest program to compute x on a universal computer. It represents the minimal amount of information required to generate x by an effective process [Kolmogorov 65].

Bennett et al. studied the definition of a universal cognitive distance [Bennett et al. '97], based on the observation that "the psychological similarity between two objects x and y is the complexity of the simplest transformation between them" [see also Chater 2003]. They start by defining the shortest program p to compute y from x:

$$K_F = \min\{l(p) : F(p, x) = y\}$$

Where $F(p,x)$ is a partial recursive function, that can be computed on a Turing machine and $l(p)$ denotes the (binary) length of vector p. To attain a symmetric definition, Benett et al. investigate reversible computations, where the overlap between the information to convert x into y and y into x is maximized. We try to define functions, by using the notion of information distance, that not only preserve complexity, but also capture the 'complexity type' or 'regularity' of an object.

K-preserving functions
A Kolmogorov complexity-preserving function g_x is a one-to-one reversible function defined from $(X, R) \rightarrow X: x' = g_x(x, k)$. Except for a limited number of special k values, where the K-complexity is reduced in x', what we call the *reduction values,* the following constraints apply for all k:

1. $x = g_x(x', -k)$.
2. $K(g_x,k|x) < K(g_x, k)$, x contains information to perform g_x.
3. The program p to compute g_x is both a minimal program, for computing x' from x and vice versa.

This last condition means that there is no extra information needed for either computation, there is maximum overlap of information and the complexity is similar in x and x'. Hence, the difference between the complexities of x and x', $K(x)$ and $K(x')$, depends only on k, not on g_x. The differences between the complexities is bounded by $|K(x) - K(x')| \leq l(k)$.
There is a fourth constraint on the implementation of the k-values:

4. $g_x(x, a.k + b.l) = a.g_x(b.g_x(x, l), k)$

Examples of K-preserving functions are translations, rotations, enlargements, repetitions, polynomials, permutations,... On a circle, translations in each direction and resizing preserve the complexity. An ellipse can additionally be stretched along both its axes, however, for a special reduction value k, the ellipse becomes a circle, that is of lower complexity.

Set with the same regularities
By applying all existing g_x functions on x by varying the k values, we become a set C_x of elements with the same regularity. Each element of this set got the same K-preserving functions and will generate the same set (following from constraint 4). This means that they all share the same *regularity*.
There are dependencies among the K-preserving functions. If for a certain x: $g_{x,1}(x, k_1) = g_{x,3}(g_{x,2}(x, k_2), k_3)$, (with all k non-reduction values), $g_{x,1}$ can be composed out of $g_{x,2}$ and $g_{x,3}$ for *all* elements of C_x. After elimination of the dependent $g_{x,i}$, a set of independent K-preserving functions $C_g\{g_{x,i}\}$ remains, which represents the *degrees of freedom* of the regularity. The set C_x got Πk_i elements and each element in the set is represented by its $(k_1, k_2, \ldots)$, after having

chosen a reference element (with all k's set to 1).

Because all elements of C_x share the same regularities and are at close (universal cognitive) distance of each other, this set has its importance in *unsupervised concept learning*. The set can be learned from a few examples. A random element or 'typical example' x of a target concept set c generates a set $C_{x,}$ by finding a set of independent K-preserving functions. If the example has all regularities of the concept, c will be a subset of C_x (see later). Only when the element would be a reduction value, C_x won't cover all elements and if the target concept set c doesn't correspond with the concept C_x (no subset or superset), x contains not enough information of $c \backslash C_x$ and that part cannot be learned from x.

Components

A minimum description of a vector x can be associated with a set $C_x(x)$ of independent K-preserving functions. The object x can then be associated with its k-values (k_1, k_2, …). This information is not necessarily presented explicitly in the minimal code of x. However, all elements of C_x have the same average K, but their must codes be different, so this information must be present at least implicitly. The minimal program p that takes (k_1, k_2, …) as input and writes the correct x of C_x, is smaller than K(x), because all information on the k-values can be removed. This program corresponds with what Rissanen called 'the summarizing property' of data in its two-part code [Rissanen '99], where the first part describes the set A where the data belongs along with other instances that share the same property and where the second part describes x_n in A.

On the other hand, the two-part code can be interpreted as a separation of the random from the structural part. There is a clear distinction between the data that is accounted for the ordered part and the data that is not described, the apparently random part [eg. Crutchfield 93]. Together with the here developed separation of the regularity p and the characteristics k_i, we can write the minimal description code as a *three-part code***.**

Learning

Concept learning is the description of the concept set [Solomonoff 99]. This can be seen as the union set of all $C_{x,j}$ of the training vectors $x_{j,}$ with additional constraints on the characteristics k_i of the set. The first part describes the regularity, it serves for the induction, and the second part restricts this set.

Most learning algorithms expect the learning process to start from a 'good description' of the observations, meaning a compressed description of the data. We argue that if the target concept indeed corresponds with a subset of $\cup C_{x,j}$, non-compressed descriptions generate difficulties in the learning process. The main argument is that the hamming distance changes drastically between elements of C_x, neighbors according to the information distance are not necessary neighbors in the euclidean space [Bennet '97].

Learning algorithms that use a distance metric (like the hamming distance), as in case-based reasoning or nearest-neighbor learning, expect instances of the concept to be 'close'. These learning algorithms will thus have to find the description according to the K-preserving functions. Moreover, we believe that errors will have to be defined in terms of this description to attain a 'universal cognitive distance'.

The same problem occurs in learning algorithms that consist of combinations of constraints on the input vector, like boolean expressions, decision trees, neural networks or descriptions on attribute-value pairs. The elements of C_x can be highly different input vectors, causing a lot of redundancy in the constraints to be learned. These learning algorithms define concepts by regions in the euclidian space [Blumer 89]. However, instances are not necessary close to each other in the euclidean space, so the learning can generate complex, redundant descriptions. Consider for example, the learning by a boolean expression of the set of the boolean vectors with an equal number of true and false values.

To solve this problem, the learning of visual concepts (by neural networks, etc.) typically starts with translating and resizing the image to a reference position and size, by calculating its centre of gravity and size. However, this doesn't work for compound objects, where the minimal description of the whole is the sum of the minimal descriptions of the objects. It is necessary to separate the objects, the learning algorithm should recognize the individual objects.

Once the input is written by its (k_1, k_2, …), constraints on *this* inputvector can learn to recognize the target concept c out of C_x. A similar approach is principal component analysis [Hyvarinen 99] and kernel-based algorithms, where the input is transformed into components to attain a more compact description that better describes the relevant characteristics.

"A fundamental problem in learning and reasoning is finding the right representation… a representation in terms of its salient features would greatly facilitate recognition" [Zemel 93, also in Hyvarinen 99], and "defining those features for a broad class of objects is a very difficult problem ".
The here developed analysis suggests a concept-dependent representation, where the representation captures the regularities of the concept.

References

[Barlow 2000] Barlow H.B., "Redundancy reduction revisited". Network: Computation in Neural Systems, 2001.

[Bennett 97] Bennett C.H., Gacs P., Li M., Vitanyi B. and Zurek W.H. "Information Distance". IEEE Transactions on Information Theory, Vol. 44, No. 4, july 1998.

[Chater 2003] Chater, N. and Vitanyi, P., "Simplicity: A unifying principle in cognitive science?" Trends in Cognitive Sciences, 7:1, 19--22 , 2003.

[Crutchfield '93] J. P. Crutchfield, "The Calculi of Emergence: Computation, Dynamics, and Induction", Physica D 75, 11-54.,1994.

[Hyvarinen '99] A. Hyvärinen. "Survey on Independent Component Analysis". Neural Computing Surveys 2:94--128, 1999.

[Kolmogorov 65] Kolmogorov, A.N., "Three approaches to the definition of the concept 'quantity of information'", Problems in Information Transmission, 1 (1):1-7, 1965.

[Rissanen 89] Rissanen, Jorma, Stochastic Complexity in Statistical Inquiry. World Scientific, 1989.

[Rissanen 99] Rissanen, J. "Hypothesis Selection and Testing by the MDL Principle.", in The Computer Journal , Vol. 42, No, 4, 1999.

[Solomonoff 99] Solomonoff, Ray. "Two Kinds of Probabilistic Induction", in The Computer Journal , Vol. 42, No, 4, 1999.

[Zemel '93] Zemel, Richard S. "A Minimum Description Length Framework for Unsupervised Learning", PhD thesis, University of Toronto, Canada, 1993.