Empirical Evaluation of the Effects of Concept Complexity on Generalization Error

Roberto Esposito [1]

Abstract. In this paper we focus on the relationship between concept complexity and generalization error of learned concept descriptions. After introducing the concept of *compressibility*, we suggest how it could be usefully exploited in order to estimate from the training data the (Kolmogorov) complexity of the concept to be learned. Then, we present an empirical apparatus which allows us to study the relationship between the estimated target concept complexity and the generalization error of different learning algorithms. Results show a linear relationship between the two variates: the generalization error appears to increase as the target concept becomes more complex. While this is expected, quite interesting is the fact that the relationship seems to be (only) linear. Moreover, while the degree of correlation changes for different learners, the "linear" relationship seems not to be affected by the particular learning algorithm.

1 INTRODUCTION

It is widely believed, in science, that more simple theories are bound to be more predictive/explicative (see, for example, [3] for a discussion). Even though appealing to the intuition, this idea encounters severe difficulties in the very definition of what "simple" is.

An attractive approach to quantifying complexity is to consider *algorithmic complexity* [6]. Applying this notion to a concept, a program corresponds to a concept description, and reconstruction amounts to correctly predicting instances.

Unfortunately, Kolmogorov's complexity is not computable, even though computable extensions of it do exist [7]. Nonetheless, Kolmogorov's formal definition can be exploited to introduce a less elegant, but more practical, notion of complexity, namely, compressibility.

2 ESTIMATION OF CONCEPT COMPLEXITY

Let us define a concept $\mathcal{C}$ as a subset of a metric space $\mathcal{X}$. We will call *positives examples of* $\mathcal{C}$ the points in $\mathcal{C}$ and *negatives examples* the points in $\mathcal{X} - \mathcal{C}$. In this work, we will focus on spaces containing a finite number of points.

Let us consider a universal machine U and denote with Π the set of syntactically correct programs for U.

The notion of compressibility can be derived from Kolmogorov complexity by recognizing that the algorithms in Π can be, in general, decomposed into two parts: an algorithmic part G that encodes the logic of the program, and the information D, which is manipulated by G. We can hence write Kolmogorov complexity as:

$$K(x) = \min_{\pi \equiv \langle \mathsf{G}, \mathsf{D} \rangle \in \Pi : U(\pi) = x} l(\pi)$$

We say that G is a decompression algorithm for $\mathcal{C}$, and that D is a compressed representation of $\mathcal{C}$. We try to find a pair $\langle \mathsf{G}', \mathsf{D}' \rangle$ which is "small" with respect to $\mathcal{C}$, even though it is not necessarily the smallest one. We do this by fixing the compression algorithm (and hence, also G), and estimating concept complexity by measuring the size of its compressed representation.

Unfortunately, it is not possible to compute exactly concept compressibility without having the whole concept at hand. As a consequence, we will make a simplifying step: we shall estimate its compressibility on a learning set.

2.1 COMPRESSION ALGORITHMS

The first compression algorithm we tested is based on a greedy covering procedure. We say that a set S of spheres, defined over a finite metric space $\mathcal{X}$, is a *cover* of a concept $\mathcal{C}$ if any point that belongs to the concept belongs to one of the sphere in S and vice versa.

Let us denote by $S_0(\mathcal{C})$ the smallest cover of $\mathcal{C}$ and by n the dimensionality of $\mathcal{X}$.

The estimate of concept compressibility given by the optimal covering algorithm is set to be:

$$\gamma_{\mathbf{COV}} = \frac{|S_0(\mathcal{C})|}{|\mathcal{X}|} \tag{1}$$

Since the optimal covering algorithm is very computationally demanding and since we assume to not know the target concept in advance, we cannot use the formula given above as is. Instead, we make use of a greedy covering algorithm and run it over the learning set. Let us denote the learning set by L and by $\mathcal{H}_{\mathbf{GCA}}(L)$ the cover found by the greedy covering algorithm (**GCA**) when run on L; the formula used to estimate the concept compressibility is:

$$\gamma_{\mathbf{GCA}} = \frac{|\mathcal{H}_{\mathbf{GCA}}(L)|}{|L|}$$

The second compression schema we used is a file zipping program, one of the standard tools available on any modern operating system. In particular, we used the freely available "gzip" program, which has already proved to be a valuable tool for the estimation of Kolmogorov complexity from data [1]. Let us denote with $F(L)$ the representation of L on file. The compressibility of the target concept is estimated by measuring the ratio:

$$\gamma_{\mathbf{ZIP}} = \frac{\|\mathbf{ZIP}(F(L))\|}{\|F(L)\|}$$

[1] Universitá di Torino, Dipartimento di Informatica, C.so Svizzera 185, 10149 Torino, Italy

Where **ZIP**$(F(L))$ is the compressed version of $F(L)$, and with the notation $\|\cdot\|$ we denote the size, in bytes, of the enclosed file.

It is worth noticing how much the two compression algorithms differ. In fact, even though both of them are meant to approximate Kolmogorov complexity, they exploit very different characteristic of the data. In order to make the difference between the two algorithms evident, let us consider the set of even integers less than 100 and denote it with $\mathcal{C}_2$. $\mathcal{C}_2$ admits $2j$ $(1 \leq j \leq 49)$ as a very simple description (and hence $K(\mathcal{C}_2)$ is bound to be small), but it cannot be compressed at all by **GCA**. However, it is easy to verify that the task of compressing $\mathcal{C}_2$ is an easy one for **ZIP**.

3 EXPERIMENTAL SETTING

We generated a large number of two dimensional target concepts of varying complexity. The complexity parameter were controlled by varying then number of different shapes included in the concept. These concepts have been used to acquire both an estimation of one of the compression measures γ, and an estimation of the generalization error ω of a number of learning algorithms. The error ω has been computed on independent test sets.

Given a learning algorithm $\mathcal{A}$, and a compression algorithm $\mathcal{Z}$, we performed 1000 experiments, each of which ended up with a pair (γ, ω). The following learning algorithms were tested: a simplified version of **CART** [2], **k-nearest neighbors** [4], **GCA**, and **AdaBoost** [5] along with two different weak learners (a sphere inducer and a decision stump inducer).

A useful way to think to the experiments is the following. On one hand we have a sampling of the extension of the target concept (the learning set), on the other hand we have a guess about its intension (the learned hypothesis). The experiments study the relation between the complexity of the concept extension and how good is the guess about its intension.

4 EXPERIMENTAL RESULTS

The experiments show a linear relationship between the generalization error of the learning algorithms and the compression estimate provided by the two measures we described. Table 1 reports the correlation coefficients of the two variates for all the experiments we performed.For the sake of illustration, we report in Figure 1 the scatter plot for the case **GCA**/$\gamma_{\mathbf{GCA}}$.

Table 1. Correlation coefficients for learning algorithm/compression estimator pairs.

Algorithm	$\gamma_{\mathbf{GCA}}$	$\gamma_{\mathbf{ZIP}}$
AdaBoost + **GCA**	0.75997	0.21943
AdaBoost + **SP**	0.55294	0.57153
GCA	0.75913	0.32898
CART	0.56207	0.28868
KNN	0.72997	0.31174

As it appears, the results concerning $\gamma_{\mathbf{ZIP}}$ estimation are less encouraging. The **ZIP** algorithm, in fact, seems to be less sensible to the variations in the complexity of the concepts underlying the training sets. This was not totally unexpected since the choice of the compression algorithm is an important parameter and different compressors are likely to behave differently on a fixed domain. For the domain presented in this paper in particular, **GCA** is a natural complexity estimator candidate, while **ZIP** is not.

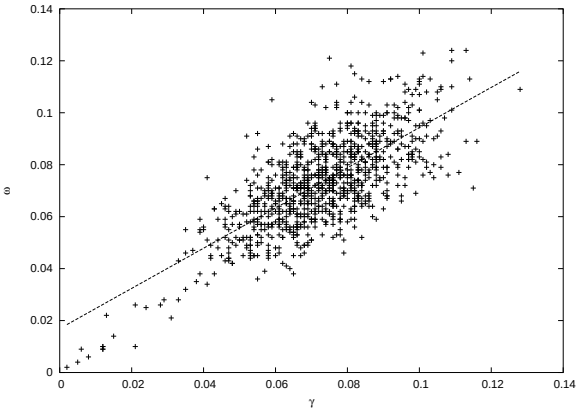

Figure 1. Generalization error of the **KNN** algorithm vs $\gamma_{\mathbf{GCA}}$.

5 CONCLUSIONS

In this paper we presented an empirical apparatus which allowed us to study the relation between the error of different learning algorithms and two different compressibility definitions. The outcomes show a linear correlation between the two variates, namely, between compressibility and generalization error.

The experiments show that the correlation is notably stronger for one of the two compression estimates; there is thus evidence that the choice of the particular compression algorithm is a delicate one. Anyway, it is interesting to observe that a positive correlation is observed through all experiments, which tested very different learning algorithms. As a consequence, the compression estimates capture some important characteristics of the training data, which seem to be relevant for the induction problem in general.

A possible explanation of this phenomenon is that the compression measures are actually estimating the complexity of the underlying concept. In other words, they may be exploiting the learning data in order to guess if the concept to be learned is a difficult one.

Following this intuition, we provided a definition of compressibility in terms of Kolmogorov complexity. This definition seems to be an interesting one. In fact, on the one hand it exposes the connection between compressibility and complexity and, on the other, it nicely explains the difference in behavior of the two compressibility measures we tested in practice.

REFERENCES

[1] Dario Benedetto, Emanuele Caglioti, and Vittorio Loreto, 'Language trees and zipping', *Physical Review Letters*, **88**, (2002).

[2] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Wadsworth, Belmont, CA, 1984.

[3] P. Domingos, 'Occam's two razors: the sharp and the blunt', in *Proc. 4 th Int Conf Knowledge Discovery and Data Mining*, pp. 37–43. AAAI Press, (1998).

[4] R. Duda and P. Hart, *Pattern Classi cation and Scene Analysis*, Wiley, 1973.

[5] Yoav Freund and Robert E. Schapire, 'Experiments with a new boosting algorithm', in *Proc. 13th International Conference on Machine Learning*, pp. 148–146. Morgan Kaufmann, (1996).

[6] M. Li and P. Vitányi. An introduction to kolmogorov complexity and its applications, 1993.

[7] Jurgen Schmidhuber, 'Discovering solutions with low kolmogorov complexity and high generalization capability', in *International Conference on Machine Learning*, pp. 488–496, (1995).