Universidade de Aveiro

Departamento de Electrónica, Telecomunicações e Informática

Algorithmic Information Theory (2021/22)

Lab work nº 2 — Due: 20 Dec 2021

1 Introduction

Consider the problem of determining the "similarity" between a target text, t, and some reference texts, r_i. For example, each r_i could be a sample text from a certain language and t could be a text whose language needs to be determined.

Usually, the traditional approach to solve this *classification problem* begins with feature extraction and selection operations. The features obtained are then fed into a function that maps the feature space onto the set of classes and performs the classification. One of the most difficult parts of this problem is how to choose the smallest set of features that retains enough discriminant power to tackle the problem.

The representation of the original data by a small set of features can be seen as a special form of *lossy data compression.* This suggests a question: Can data compression be explicitly used to approach classification problems, removing the need for a separate feature extraction stage? The answer is affirmative, i.e., it is possible to adopt an information-theoretic approach to the classification problem, bypassing the feature extraction and selection stages. In other words, compression algorithms can be used to measure similarity between files.

The idea is the following. For each class, represented by the reference text r_i, we create a model that is a good description of r_i. By a "good description", we mean a model that requires fewer bits to describe r_i than the other models, or, in other words, that is a good compression model for the "members of the class" r_i. Then, we assign to t the class corresponding to the model that requires less bits to describe it, i.e., to compress t.

2 Work to be done

1. Develop a program, named `lang`, that accepts two files: one, with a text representing the class r_i (for example, representing a certain language); the other, with the text under analysis, t. Modeling should be performed using the finite-context models studied in the previous Lab Work. Other parameters, such as the order of the context model and the parameter α of the probability estimator, should also be provided to the program. The program should report the estimated number of bits required to compress t, using the model computed from r_i.

2. Based on the `lang` program, build a language recognition system, `findlang`, that, from a set of examples from several languages (the r_i), provides a guess for the language in which a text t was written.

3. You should obtain results with as many different languages as possible (definitely, more than ten). It is up to you to find texts that are good representatives of the language and test them with appropriate examples (a good source of these texts can be found in `https://sourceforge.net/projects/la-strings/files/Language-Data/`).

4. Develop an application, `locatelang`, that can process a text containing segments written in different languages. This application should return the character position at which each segment starts, as well as the language in which the segment is written.

5. As a bonus challenge, explore the possibility of using combinations of several finite-context models (with different orders) to represent each language.

6. Elaborate a report, where you describe all the steps and decisions taken during the development of the work, and include relevant and illustrative results that were obtained. Show also results regarding the classification accuracy of the system as a function both of the length of the references and of the length of the target texts. This report should be convincing enough for motivating someone to “buy” your product!