universidade de aveiro **deti** departamento de electrónica, telecomunicações e informática

Information and Coding (2021/22)

Project #03

Finite Context Models

Deadline: 31st January 2022

- All the programs should be implemented in C++.
- Document all the code using the Doxygen tool.
- Create a Github project to manage the several versions of your software.
- Each program should be demonstrated during the classes

Part A

Usually, the purpose of studying data compression algorithms is twofold. The need for efficient storage and transmission of data is often the main motivation. However, underlying every compression technique, there is a model that tries to reproduce as closely as possible the information source to be compressed. This model may

compression, we use a specific type, called discrete-time Markov chain or finite-context model.

A finite-context model can be used to collect high-order statistical information of an information source, assigning a probability estimate to the symbols of the alphabet, according to a conditioning context computed over a finite and fixed number of past outcomes. Our main goal is to predict the next outcome of the source. To do this, we infer a model based on the observed past of the sequence of outcomes.

1. Develop a program, named **fcm**, with the aim of collecting statistical information about texts, using finite-context models. The order of the model, k, as well as the smoothing parameter, α, should be parameters passed to the program. This program should provide the entropy of the text, as estimated by the model.

Part B

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

Project #03

Atualização automática a cada 5 minutos

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`.

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 implemented in Part A. 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

Project #03

several languages (the r_1), 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. As a bonus challenge, explore the possibility of using combinations of several finite-context models (with different orders) to represent each language.

Part C

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!.

You should include the Doxygen documentation (if available) and a link to your repository.

Project #03

Atualização automática a cada 5 minutos