Machine Learning 2021 Course — Homework

Matteo Matteucci
matteo.matteucci@polimi.it

Marco Cannici
marco.cannici@polimi.it

Politecnico di Milano — July 3, 2021

Predicting taxonomic identity and genetic composition based on codon usage bias levels

Background

> «The coding DNA of a genome describes the proteins of the organism in terms of 64 different codons that map to 21 different amino acids and a stop signal. Different organisms differ not only in the amino acid sequences of their proteins, but also in the extents in which they use the synonymous codons for different amino acids. The inherent redundancy of the genetic code allows the same amino acid to be specified by one to five different codons so that there are, in principle, many different nucleic acids to describe the primary structure of a given protein. Coding DNA sequences thus can carry information beyond that needed for encoding amino acid sequence. Thus, one may ask: is it possible to classify some properties of nucleic acids from the usages of different synonymous codons rather than, with much greater computational effort, from individual nucleotide sequences themselves?
>
> This data set enables a preliminary analysis on this topic. In particular, codon usage frequencies from several organisms are studied to identify if they can be used to classify codon usage (i) in terms of viral, phageal, bacterial, archaeal, and eukaryotic lineage, as well as (ii) classifying codon usage by cellular compartment (nuclear, mitochondrial, and chloroplast DNA in their respective organelles). »
>
> — Khomtchouk, Bohdan B. "Codon usage bias levels predict taxonomic identity and genetic composition." bioRxiv (2020).

Dataset Composition

The data set consists of a total of 13028 organisms (samples) classified both based on the organism species (11 classes) and on the DNA type (11 classes). Features are all real valued and no data is missing, except for the `AGA` and `ACA` codon frequencies on test samples.

Features

Every sample is described by 67 attributes:

- `SpeciesName` : the species of the organism
- `SpeciesID` : an identifier of the species
- `Ncodons` : the total number of codons in the entry
- `UUU` - `AUG` : the 64 codon frequencies (the codon occurrency count divided by the total number of codons `Ncodons` in the entry)

Classes

The dataset features two classification problems. In the first, the goal is to classify samples in 11 different classes based on the value of the attribute `Kingdom`

- `Kingdom` (11 classes): a three letter code referring to the organism species. In particular, identifiers are: 'arc' (archaea), 'bct' (bacteria), 'phg' (bacteriophage), 'plm' (plasmid), 'pln' (plant), 'inv' (invertebrate), 'vrt' (vertebrate), 'mam' (mammal), 'rod' (rodent), 'pri' (primate), and 'vrl' (virus)

The second tasks is similar, but samples are classified based on the DNA type:

- `DNAtype` (11 classes): an identifier referring to the type of DNA sample: 0 (genomic), 1 (mitochondrial), 2 (chloroplast), 3 (cyanelle), 4 (plastid), 5 (nucleomorph), 6 (secondary endosymbiont), 7 (chromoplast), 8 (leucoplast), 9 (NA), 10 (proplastid), 11 (apicoplas), and 12 (kinetoplast)

Train/Test splits

The dataset is provided already split into train (10422 samples) and test (2606 samples) sets. While the training set comes with all the 64 codon frequencies for each sample, a malfunctioning in the analysis procedure caused the measurements of the `AGA` and `ACA` codon frequencies to be lost in all test samples. Features are all real valued and no data is missing, except for the `AGA` and `ACA` frequencies on test samples.

- `train.csv` : 10422×69 dataset
- `test.csv` : 2606×67 dataset

Requests

1. Perform a **preliminary analysis** on the data. For instance, but not limited to, visualize samples, identify if features (i.e., codon frequencies) are correlated, determine which are most correlated with the target classes, and inspect the distribution of samples among classes. Using **clustering**, study if there are structures in the data that allow samples from different classes (both DNA type and Kingdom) to be easily identified. Compare the performance of different clustering algorithms and distance measures using the metrics presented during the course.
2. Perform **classification** in order to classify organisms into the 11 `Kingdom` classes. Perform features selection, compare different algorithms and identify the one that works the best on this dataset. Finally test the performance of the best algorithm on the provided test set.
3. We want to recover from the data loss of the `AGA` and `ACA` frequencies on test samples. Train a **regressor** to predict the values of the `AGA` and `ACA` features given the remaining ones. Compare different regression algorithms for this task. Since the `AGA` and `ACA` features are missing in test samples, use only the training data for this step and make use of robust evaluation techniques to compare algorithms. You can either use two separate regressors to predict each missing feature, or a single one that predicts both (see notes).
4. Use the regression model trained at the previous step to recover the `AGA` and `ACA` codon frequencies, by predicting their value of each test sample. Determine if the test performance of the best model found at step (2.) improves if the `AGA` and `ACA` frequency values are also used for prediction.

Note: In Scikit-Learn, multi-value regression can be performed as usual by providing a target value `y` with shape `[N_samples, N_target_values]`. The regressor will output a prediction with the same shape as `y`, in this case `[N_samples, 2]`, with one column for each target value predicted.

Note: As the data set contains several samples, some algorithms and visualizations may be particularly slow to run. You can opt for selecting a smaller portion of the data to perform part of the analysis (part of the preliminary analysis may be devoted to select a meaningful portion on the data set), or other techniques that you think could be useful to mitigate the issue.

Submission

Use this form to submit your homework: `http://tiny.cc/ML2021Submission`
The deadline is July 31 at 23:59 CEST (Rome)

- Create a Jupyter notebook to answer all the requests, using the libraries presented during the laboratory classes.
- Include Name, Surname and Student ID in the notebook.
- You are free to use the structure that you prefer within the notebook. However, please use markdown cells (`Cell > Cell Type > Markdown`) to insert section titles and clearly identify the different requests. You are free to add subsections to make the notebook more readable.
- Add text cells (markdown) to briefly explain what you did and why, and to help you answer the requests.
- Please, check that the notebook can execute correctly before submitting your work.
  Press `Kernel > Restart & Run All` and check that all cells execute correctly without errors.
- The output of notebook cells should be included in the submitted notebook.

Evaluation

The project evaluation is not performed based on the (clustering, classification, regression) scores you obtain in each request, but rather on the soundness of your analysis and the choices you made to solve issues (if any) you may encounter. A good project follows the evaluations discussed during the lectures and lab sessions, while also adapting and integrating the analysis based on the project.

The following is the evaluation criterion used in the previous year's project. Notice that it may slightly vary in this year evaluation, but you can use it as a rough reference.

Type	Points	Comment
Penalty	0	Does compile with no error
	-0.5	Does not compile, but it is an easy fix to make it work
	-1	Does not compile, and it would require more that 10 mins of debugging the errors
Preliminaries	0	Completely wrong, or one key concept is not taken into account
Classification	1	Ok, but poorely discussed. E.g., just a copy-paste of labs with no comments, or some conceptual errors
Regression	2	Well done, good comments and analysis, adapted code/analysis, he/she interpreted the results