Challenge Data: Technical specifications

Introduction

This document aims at facilitating the submission of a challenge by precisely specifying the required pieces of information. After a sketch of how the challenge operates, this document details and provides remarks on some of the most important aspects of challenge creation. At the end, one can find an example of specifications.

General organisation of the challenge

Challenge Data consists of statistical learning problems of types such as regression, classification, ranking, and prediction, etc. In each case, it is about "learning" a function $f : x \mapsto y$ from a set of examples (couples (x, y)), called training set. The quality of the learning is measured from predictions $\hat{f}(x)$ on a set of examples x such that the solution $f(x)$ is not given to participants, called test set. These submitted results are compared automatically based on a predefined metric with the ground-truth, which is stored on the platform `https://challengedata.ens.fr` and is not accessible to participants. The obtained score, which is the only thing revealed to participants, allows to classify the different contributions with no doubt on the probity of participants.

Problem definition

One of the most important element when creating a challenge is the definition of the problem. This consists in statistically learning a function f from examples. For binary classification problem, we have for example $f(x) \in \{0, 1\}$; for the problem of univariate regression, $f(x) \in \mathbb{R}$. In other words, it is about constructing a function $\hat{f}$ such that $\hat{f}(x_i) \approx f(x_i)$ for all training samples x_i, and which generalises (predicts equally well) over the test set. In general, the input x is constructed as a Cartesian product of different variables, called “features”, which can be either continuous-valued (intensity of a pixel in colour, spatial position, etc), or discrete-valued (the gender of an individual, etc). It is important to note that the information contained in x needs to be sufficient to obtain a prediction $\hat{f}(x)$.

One obstacle to avoid in the problem construction is that x contains all or part of $f(x)$ (without any linear or non-linear transformation), or eventually of another $f(z)$. This occurs frequently in time-series. In fact, in order to predict the future values in a time-series $(s_t)_t$, a simple approach is to define the vectors in the intervals of a certain length $x_t = (s_{t-p}, \dots, s_t)$ and to fix $f(x_t) = s_{t+1}$. In this case, $f(x_t)$ is in the vector x_{t+1}, without any need to learn anything. Even when assuming that the order of vectors have been changed by a random permutation, whenever $p \geq 1$, it is possible to find out the consecutive (in-time) vector by comparing the last p coordinates with the first p coordinates of the other vectors. Such kind of challenge is ill-posed and needs to be avoided.

Data

The data are divided under two criteria, forming four distinct sets: training or test, input or output. All the training data $(x_i, f(x_i))_i$ and the test inputs $(x_k)_k$ are provided to participants, while the test outputs are stored on the platform, which permits to evaluate the submissions.

Every couple $(x, f(x))$ needs to be associated with a unique identification number, called index and written

“ID”. In particular, there should be no intersection between the indices of the training and test sets. As the inputs x and outputs $f(x)$ are split between two files, it is compulsory to guarantee the equality of the IDs between these files. It is recommended to use increasing and consecutive indices across the training and test sets, for instances with $0, \ldots, N_{tr} - 1$ in the training set of size N_{tr} and $N_{tr}, \ldots, N_{tr} + N_{te} - 1$ in the test set of size N_{te}.

For statistical reasons, the number of test data should not be too small, at least larger than 1000 preferably. In fact, within the platform, the test set is randomly divided into two partitions of same size, one called public, the other called private. Only the results obtained from the public partition will be communicated to participants during the competition; one thus needs to ensure that the measure does not fluctuate too much on half of the test set.

In case of numerical, categorical or text data, the CSV format shall be favoured for its simplicity and its interoperability. Each file needs to contain a header line indicating the title of each column (possibly some description), the first column containing the index “ID”. The index column must be identical between corresponding input and output files. Moreover, the header line should be identical for all input files.

Even if the input data is not adapted to the CSV format (for example in case of images), this format shall necessarily be used for the output data.

While missing input data can make the game more interesting, as long as their overall quantity remains reasonable, it is imperative that no output data are missing so as to permit a correct evaluation of the score.

The separator should be commas (,) for technical reasons.

An input CSV file shall thus have the following format:

ID, Feature1, Feature2, ...
0, 1.215, ’Joe’, ...
1, -785.5, ’Jack’, ...
...
987, 0.343, ’William’, ...

In the end, four files are to be provided:

Training data :
1. Input in format .csv or .zip : input_training.csv/.zip
2. Output in format .csv : output_training.csv
Test data:
3. Input in file .csv or .zip : input_test.csv/.zip
4. Output in format.csv : output_test.csv

A .zip file containing additional information (such as a python notebook) can optionally be uploaded.

In total, it is preferable that the data size be smaller than several GBs, so that is is possible to participate in the challenge with a standard computer. In all cases, it is necessary to make the size of the output test set smaller than 10 MBs.

Metric

The metric associates to each submission a numerical score permitting to judge its value. The simplest example for a classification problem is the classification accuracy (the ratio between the number of entries assigned to the good class and the number of entries of the test set), and the quadratic error for regression problems. As always, a metric is not necessarily a raw performance measure and it can integrate more refined criteria, such as the robustness measured by the AUC curve.

The metric needs to be chosen carefully according to the nature of the data and the scientific and technical objectives. It cannot be modified once the challenge has started.

Most of the standard metrics are already incorporated on the platform. It is nevertheless possible to add some new metrics by providing an appropriate python script.

In order to be compatible with the automatic and random separation of the test set into a private part and a public part, the metric needs to be separable with respect to the individual examples. More precisely, if $\mathbf{y}, \hat{\mathbf{y}} \in \mathbb{R}^{N_{te} \times p}$ are the matrices respectively containing the true outputs and the outputs provided by the users for the test set, where each line y_i contains an example and each coordinate a value to predict, then the metric needs to be written as:

$$L(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N_{te}} \sum_{i=1}^{N_{te}} \ell(y_i, \hat{y}_i)$$

In this way, this formula can be restricted without any ambiguity to the private or the public part.

Calibration

Before opening the challenge to the competition, it is important to calibrate it by testing standard algorithms. The score obtained from a company's submission is called "benchmark" on the platform, and appears as such in the ranking. This benchmark permits to verify that the challenge does not contain any obvious flaw and to guide students to the type of performance that they should attain to be in the game.

For this benchmark to be visible in the ranking of the challenge, it needs to be submitted from the challenge provider's admin interface. The submission process can be repeated as often as necessary: the last benchmark submission then becomes the public one.

Overfitting

In order to avoid overfitting training data, it is suggested to verify the absence of duplicates in the training dataset. At the level of the platform, the number of submissions is limited to two per participant (or per team when applicable) within 24 hours. In the end, only the score calculated over the public part of the test data, called public score, is revealed to participants when they submit. The score computed on the private part of the test set, called private score, is only revealed to participants twice: at an intermediate academic date, and at the end of the challenge.

Challenge example

Problem description

The goal of the challenge is to predict for multiple American stocks which volume will be exchanged during a certain period. Since the American stock market opens at 9:30am and closes at 4pm, the competitors will have access to the transactions between 9:30am and 2pm and need to predict the transactions between 2pm and 4pm.

Data description

The input data contains the exchanged volume (in dollars) of a certain set of stocks and aggregated dates over some periods of 5 minutes. Each line is defined by a unique "ID" and corresponds to a certain day (defined by "date") and to a certain stock (defined by "product_id"). The exchanged volumes are the sum of the values exchanged on the market and are labelled with the time of the beginning of the aggregation (from 9:30am to 1:55pm).

The first line of the input file contains the header, and the columns are separated by commas. The three first corresponding columns are:
- the "ID": the identification number, it is linked to the ID of the output file,
- the "date": related to a certain day, shared by some stocks (for practical reasons, the days are mixed randomly),
- the "product_id": the identification number of the stock, it is related to a particular company (the stock #236 corresponds to the same company in the training and test file).

The remaining columns are the volumes exchanged in dollars over the period of 5 minutes. The time labels are provided in military format. For example, the column labelled with 10:00:00 corresponds to the volume exchanged on the market between 10am and 10:05am.

Here is an example of an input file:

```
ID,date,product_id,09:30:00,09:35:00,...,13:55:00
0,1,236,121984.4,644041.2,...,38941.0
1,1,239,2594.3,2587.1,...,964.2
2,1,254,96348.1,73186.9,...,65156.9
```

The training output file contains the output for each "ID", which shows the volume exchanged for the same stock and the same day between 2pm and 4pm. The first line of the file contains the header and columns are separated by commas:

```
ID,TARGET
0,11887952.0
1,770918.0
2,3004214.0
```

Thanks to the file test_input, participants need to provide a test output file in the same format as the training output file (associating each ID with the predicted volume during 2pm-4pm)

Metric

The metric used in this challenge to rank participants is the average absolute relative error : $\frac{1}{N}\sum_{i=1}^{N}\left|1-\frac{\hat{f}(x_i)}{f(x_i)}\right|$.