INGI2262 – Machine Learning

Assignment 5

A Machine Learning Competition

1 Introduction

This final (and longer) project will give you the opportunity to have some fun while competing with other students to solve a real and difficult prediction task. Your job will be twofold.

1. build the best possible classification model on a training set to predict the (undisclosed) class labels on a given test set,
2. predict the actual classification performance your model will have on the test set.

1.1 Design choices

The "training" set is referred here in a broad sense, that is the fraction of the dataset on which the class labels are disclosed. It is up to you to decide what to do with this labeled set and, for instance, whether you split it (once or several times, possibly recursively) into actual training versus some validation fraction. More generally, this project is **intended to be open** as it is the case for a real task. Many design choices are left open and **you** must specify them. Here is a non-exhaustive list of things you might have to consider.

- Do you need to pre-process, filter out, normalize, ... the available data?
- How are you going to address the fact that some feature values could be numerical while others could be categorical or even binary? Are there missing values (typically represented by `NA` in R) in the training and/or the test and, if so, how to deal with such values? Are there specific feature values only observed in the test but not in the training?
- Should you consider all the available features? Should you define new or additional features from the existing ones?
- Which methodology are you going to use to learn a model, to fix its possible hyper-parameters and to predict its classification performance on the test set?
- Which learning algorithm do you consider? You are free to choose the one you estimate more appropriate to the task. It needs not be (but it might be, of course) a learning algorithm that has been presented in the course. You can even consider several learning algorithms and/or produce and combine several models as long as your final prediction defines a unique label for each test example.
  What else? ...

1.2 Competition performance metric

Since the dataset needs not be perfectly balanced in terms of class priors, the chosen test performance metric of a classifier is defined as the *balanced classification rate* (BCR). The BCR computes the classification rate for each class and reports the arithmetic average of those rates over all classes. In a binary classification context, BCR is simply the average between specificity and sensitivity:

$$BCR = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{FP+TN}\right)$$

where

- TP denotes the number of true positives on the test set,
- FN denotes the number of false negatives on the test set,
- TN denotes the number of true negatives on the test set,
- FP denotes the number of false positives on the test set.

We note that a trivial binary classifier predicting all examples as belonging to the same class has a 50 % BCR, no matter how the class priors are distributed[1], while a perfect classifier has 100 % BCR[2]. Since your task is not only to produce a model with the best possible BCR on the test set but also to predict how well your model is going to perform on this test set, we distinguish between the true test BCR of your model and your predicted $\widehat{BCR}$ on this test set.

The actual competition performance metric P (according to which you will be ranked and graded) is as follows.

$$P = BCR - \Delta(BCR) \times [1 - \exp(-\Delta(BCR)/\sigma)]$$

where

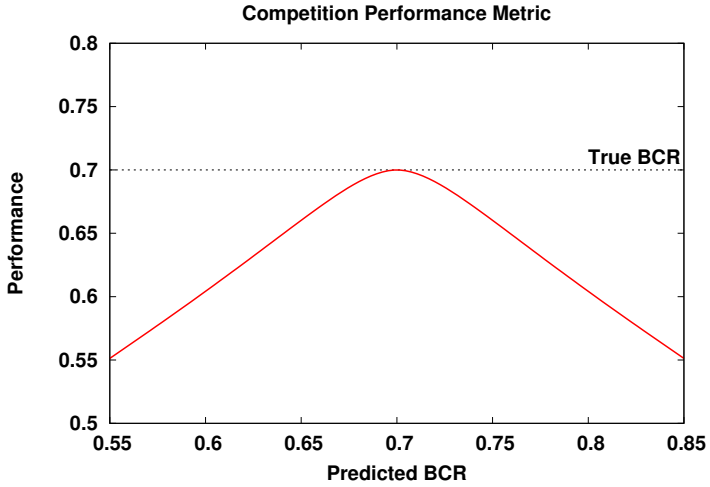
- $\Delta(BCR) = |BCR - \widehat{BCR}|$,
- $\sigma = (1/2)\sqrt{\frac{p_1(1-p_1)}{m_1} + \frac{p_2(1-p_2)}{m_2}}$, where p_1 (resp. p_2) is the fraction of correctly classified test examples of the first class (resp. of the second class) and m_1 (resp. m_2) is the number of test examples of the first class (resp. of the second class).

This competition performance metric is directly inspired from the International Performance Prediction Challenge WCCI 2006. The larger P the better. To maximize P you need to get the best possible test BCR **and** to make sure that your predicted $\widehat{BCR}$ is as close as possible to the actual test BCR of your model. Any deviation between both is penalized (by $-\Delta(BCR)$) while the influence of such penalty is limited in the region of uncertainty where $\Delta(BCR)$ is commensurate with σ, the error bar on your true test BCR.

Here is a typical plot of the performance metric P as a function of the predicted $\widehat{BCR}$ for a true test BCR being equal to 70 %.

[1] whenever the single predicted class is never actually observed in the test set, BCR = 0% assuming $\frac{0}{0} = 0$.

[2] what is the expected BCR of a classifier predicting uniformly at random among the classes? Does it depend on the class priors?



The only thing you will need to provide for us to compute P is your predicted $\widehat{BCR}$ and the predicted class labels of the test examples. From this and knowing the undisclosed true class labels, we will compute the actual BCR of your model, its estimated error bar σ and the resulting P. The highest P among all submissions will define the winner of the competition (and earn our congratulations!).

2 The Prediction Task

Personalized medicine is a recent medical model in which healthcare (prognostic, diagnostic, treatment guidance, *etc*) is tailored to each patient. It often relies on high-throughput technologies that allow to collect huge amounts of biological and medical parameters very rapidly. For instance, microarray DNA chips can measure the level of expression of thousands of genes in a single experiment and clinical variables (either continuous or categorical) can also be recorded for each patient.

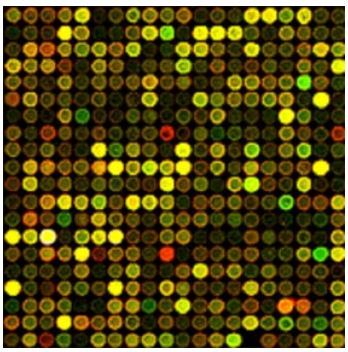


Figure 1: DNA microarray

Because of the high number of variables, such data fall in the class of 'small n, big d' problems. Indeed, the number n of samples is generally relatively small (≈ 100) compared to the number d of dimensions ($10,000$ or more). In these conditions, correct estimation of the generalization of a learned model becomes difficult. Indeed as soon as a $d \geq n - 1$, it is always possible to perfectly separate the training data (whenever such data is continuous) with a simple linear discriminant, but such a perfect performance needs not be representative of

what would be obtained on an independent sample. A proper use of machine learning methods and of evaluation protocols is thus needed to provide good generalization performances and to get a reliable estimate of such performances.

The data we are interested in this year concerns patients diagnosed with a breast cancer. Once these patients have received an initial treatment (typically through surgery and/or radiotherapy), clinicians need to know whether each patient is likely or not to suffer from a relapse, generally linked to the occurrence of metastasis of the initial tumor within 5 years after the initial treatment. Such a difficult prognosis problem is critical to choose a proper follow-up treatment. The data you will have to analyze is precisely made of a large collection of gene expression values and a collection of additional variables, such as the age of the patient, the grade of the tumor at the time of the initial diagnosis, *etc*.

The data is available on Moodle in the form of a RData file ***project.RData***. You can load it in a R session with **`load("project.RData")`**. The following objects will then be available in the environment:

- **`trainSet`** a 258×22289 data frame containing the training samples
- **`trainLabels`** a vector with the 258 corresponding class labels.
- **`testSet`** a 86×22289 data frame containing the test samples for which you don't have access to class labels.

A second file, **`Predictions.csv`** is an example of the file that you need to return. The first line is a predicted BCR value on the test set. The next lines report the predicted labels for each example in the test set, consistently with the sample Id's used as row names in the **`testSet`** data frame. This specific **`Predictions.csv`** file refers to a (fairly stupid) classifier alternating **`"Metastasis"`** and **`"No Metastasis"`** prediction on consecutive samples. Its predicted BCR (0.5) is indeed expected to be low while the predicted labels are unlikely to be correct.

Your job is to return an updated version of `Predictions.csv` under the exact same format. It is essentially generated by using the function `write.csv` on a dataframe reduced to one column and named rows.

In order to check that your updated **`Predictions.csv`** has the correct format, you can test it on https://inginious.info.ucl.ac.be/. Only the format of the file is checked, **not the actual answers**. You have to log in with your global UCLouvain account and then register to the LINGI2262 course. Passing the test on INGInious is mandatory because we will use an automated script to assess your submissions. The file, that you will eventually submit **on Moodle** (see next step), also needs to be exactly named **`Predictions.csv`** and must follow the exact same format.

3 Submission procedure

- Submit on Moodle in due time an updated version of **`Predictions.csv`**.
- Submit on Moodle a PDF file describing your design choices, the learning algorithm(s) you have been using, and any comment you consider relevant about this competition or on how you attempted to win it!

Please submit **exactly these 2 files** (one .csv file, one PDF file) and nothing else (NO archive, .rar, .gzip, .tar.gz, . . .).