# TP : classification de spam par boosting

## 1 Introduction

On dispose d'un jeu de données `spam`, créé par Hewlett Packard (HP) en 1999, qui décrit les caractéristiques de 4601 e-mails au travers de 58 variables. On le retrouve dans le paquet `kernlab`.

Les données comprennent d'une part 48 variables numériques explicatives :

— 48 variables `word_freq_WORD`, comprises entre 0 et 100, qui indiquent la fréquence en % de `WORD` parmi tous les mots ;

— 6 variables `char_freq_CHAR`, comprises entre 0 et 100, qui indiquent la fréquence en % de `CHAR` parmi tous les caractères ;

— `capital_run_length_average` est la longueur moyenne des séquences ininterrompues de caractères en majuscule ;

— `capital_run_length_longest` est la longueur maximale des séquences ininterrompues de caractères en majuscule ;

— `capital_run_length_total` est la somme des longueurs des séquences ininterrompues de caractères en majuscule (c'est-à-dire le nombre total de caractères en majuscule des le document).

— De plus, la variable `type` a deux valeurs possibles : 1 pour `spam`, 0 sinon.

On demande pour ce TP de fournir un petit rapport d'analyse préliminaire des données. Pour cela on s'appuiera sur

— le logiciel R installé sur les machines, ainsi que les paquets `adabag`, `gbm` et `xgboost` ;

— les données `spam` du paquet `kernlab`.

On donne ci-dessous une liste de questions pour guider le travail. On demandera de répondre de manière brève mais aussi précise que possible en donnant :

— le code utilisé,

— le résultat (données, graphiques),

— éventuellement un commentaire et/ou des explications sur ce que vous avez fait.

## 2 Statistiques de base

**Question 1** *Quelles est la proportion de spams ? Comment peut-on construire un classifieur trivial qui pourrait servir de référence ? Quel serait son taux d'erreur ?*

**Question 2** *Préparer pour la suite une base d'apprentissage et une base de test (par exemple 80%/20%). On s'assurera que le ratio de spam est conservé.*

## 3 Utilisation de AdaBoost

On cherche à utiliser d'abord l'algorithme AdaBoost original. On propose pour cela le paquet `adabag`.

**Question 3** *Montrer comment entraîner le modèle avec 10 itérations. Calculer le taux d'erreur sur la base d'apprentissage, puis sur la base de test. Commenter.*

**Question 4** *Tracer des courbes montrant l'évolution des erreurs sur la base d'apprentissage et de test en fonction du nombre d'itérations. On limitera la profondeur d'arbre à 3.*

## 4 Gradient boosting

On utilise ici le paquet `gbm` pour tester le gradient boosting sur les mêmes données.

**Question 5** *Montrer comment entraîner le modèle avec de 1 à 3000 itérations. Tracez les taux d'erreur d'apprentissage et de généralisation. Commenter. Vous pouvez aussi si vous le souhaitez essayer de faire varier différents paramètres, comme la profondeur des arbres et le paramètre de régularisation (shrinkage).*

## 5 XGBoost

On utilise finalement le paquet `xgboost` pour tester cet algorithme sur les mêmes données.

**Question 6** *Montrer comment entraîner le modèle avec 100 itérations. Commenter le résultat.*