INF-6422: Rapport laboratoire 2

Rendu le Mardi, 16 février 2016

à *François Labrèche*

Thomas Luinaud, Paul Berthier

Table des matières

1. Mise en contexte

1.1

Le spam, ou encore courriel indésirable, peut être défini comme étant une communication électronique non sollicitée. Selon le rapport Q1-2014 de Kaspersky, le spam représenterait aujourd'hui plus de la moitié du trafic électronique. Souvent utilisé pour des fins commerciales, le spam peut aussi être utilisé pour fins d'escroquerie ou prendre la forme d'hameçonnage (phishing en anglais) afin de tromper le destinataire dans le but d'obtenir des informations personnelles.

1.2

Bien que le coût d'envoi d'un message électronique puisse être négligeable pour les spammeurs, celui associé à sa réception peut causer des coûts non négligeables tant aux destinataires qu'aux prestataires de services compte tenu du volume élevé d'envoi. Plusieurs méthodes de détection ont par conséquent été développées afin de filtrer les messages indésirables. Une première catégorie de filtres consiste à bloquer les messages sur la base d'une liste. Cette méthode peut elle-même être divisée en plusieurs techniques, soit le recours aux listes noires (blacklisting), aux listes blanches (whitelisting), aux listes grises (greylisting), etc. Une autre approche qui a démontré son efficacité consiste à filtrer les messages sur la base de leur contenu en utilisant des méthodes d'apprentissage automatique. Le présent travail pratique vous permettra de vous familiariser avec certaines méthodes d'apprentissage automatique et d'en évaluer la performance dans un contexte de détection de spam.

2. Méthode statistique

Une approche souvent utilisée revient à filtrer les messages électroniques sur la base de leur contenu (content-based filtering). Un exemple classique consiste à filtrer les messages en fonction de la fréquence d'apparition de certains mots. Utilisez le fichier spambase afin d'appliquer une méthode statistique qui vous permettra de classifier les messages en deux catégories, soit spam (1) ou non spam (0). Compte tenu de la nature binomiale (0/1) de la variable dépendante, la régression logistique peut être utilisée comme méthode de classification. Votre variable dépendante (0/1) représente la catégorie associée (spam ou non spam) et les variables indépendantes représentent la fréquence d'apparition de certains mots.

2.1

Effectuez une régression logistique en utilisant l'ensemble des variables (57) contenues dans le fichier spambase. Divisez les données afin d'en utiliser 66% pour la phase d'apprentissage de votre modèle. À quoi servira l'autre 33% ? Donnez pour chaque variable indépendante son coefficient ainsi que son « odd ratio » associé. Quelle est la signification de ces deux valeurs ?

Les premiers 66% servent à établir notre modèle, et les derniers 33% servent à évaluer sa performance (on doit appliquer notre modèle à des données qui n'ont pas été utilisées durant la phase d'apprentissage, sinon quoi les résultats seraient biaisés).
Les coefficients et les "odd ratio" de chacune des variables sont sur le tableau 1.
En ce qui concerne les coefficients, on effectue la somme des coefficients de chaque variable multiplié par le nombre d'occurrences du mot dans le mail. Si le résultat est positif, on considère que le mail est du spam. Ainsi, plus le coefficient est élevé, plus la variable correspondante est un indicateur de spam. Au contraire, un coefficient négatif est un indicateur que le mail est sûrement légitime.
Le "odd ratio"est calculé comme $\frac{p}{1-p}$, avec p la probabilité que le mail dans lequel se trouve la variable correspondante soit du spam. Ainsi, un grand odd ratio est un indicateur de spam, et un petit odd ratio (inférieur à 1) indique au contraire que le mail est sûrement légitime.

2.2

Évaluez et discutez des performances de votre modèle en termes de : taux de vrai positif, taux de faux positif, précision, sensibilité (recall), « F-measure » et l'aire sous la courbe (ROC area). Expliquez la signification de chacune de ces mesures. Donnez la matrice de confusion de votre modèle et expliquez ce qu'elle représente.

Weka nous donne les performances de notre modèle à l'aide du tableau de la figure 1 1. Voici la signification des différentes valeurs fournies :

- **Taux de vrai positif** : C'est le taux de messages correctement classifiés comme spam.
- **Taux de faux positif** : C'est le taux de messages classifiés comme spam alors qu'ils étaient en réalité légitimes.
- **Précision** : C'est le rapport entre le nombre de messages correctement classifiés comme spam sur le nombre total de messages classifiés comme spam.
- **Sensibilité** :C'est le rapport entre le nombre de messages correctement classifiés comme spam sur le nombre de messages étant réellement du spam. C'est l'équivalent du taux de vrai positif.
- **F-measure** : C'est une formule donnant un critère de performance sous forme d'une moyenne pondérés de la précision et de la sensibilité. On la calcule de la façon suivante : $F = 2 * \frac{precision * sensibilite}{precision + sensibilite}$.
- **Aire sous la courbe** : La courbe représente le taux de vrais-positifs en fonction du taux de faux-positifs. Soient un message de spam et un message légitime choisis aléatoirement, et la question "Lequel de ces deux messages est du spam ?". L'aire sous la courbe représente alors la probabilité que notre classificateur réponde correctement à cette question.

Pour un système classificateur de mails, on peut accepter quelques vrais négatifs (du spam classé comme légitime), mais on ne veut pas de de faux positif (des messages légitimes classifiés comme spam). Le taux de faux positifs étant de 0.046 et le taux de vrais négatifs étant de 0.109, ces critères sont relativement bien respectés, mais des progrès sont à faire. Plus de 90% du spam est filtré, mais ils reste tout de même 5% de messages légitimes qui partent au spam, ce qui oblige à vérifier régulièrement sa boite spam pour voir si un message important ne s'y trouve pas.

La matrice de confusion de notre modèle se trouve à la figure 2. Les lignes représentent les deux classes (spam et légitime) correspondant à la réalité, et les colonnes les deux mêmes classes, mais pour le résultat de la classification. On peut donc retrouver tous les résultats précédents (mis à part l'air sous la courbe) à partir de cette matrice. En normalisant la matrice, plus celle-ci se rapproche d'une matrice diagonale, plus notre classificateur est performant (on n'a alors ni faux positif ni vrai négatif).

Variable	Coefficient	Odd ratio
word_freq_make	-0.3895	0.6774
word_freq_address	-0.1458	0.8644
word_freq_all	0.1141	1.1209
word_freq_3d	2.2514	9.5012
word_freq_our	0.5624	1.7549
word_freq_over	0.883	2.418
word_freq_remove	2.2785	9.7622
word_freq_internet	0.5696	1.7676
word_freq_order	0.7343	2.084
word_freq_mail	0.1275	1.1359
word_freq_receive	-0.2557	0.7744
word_freq_will	-0.1383	0.8708
word_freq_people	-0.0796	0.9235
word_freq_report	0.1447	1.1556
word_freq_addresses	1.2362	3.4424
word_freq_free	1.0386	2.8252
word_freq_business	0.9599	2.6113
word_freq_email	0.1203	1.1279
word_freq_you	0.0813	1.0847
word_freq_credit	1.0474	2.8503
word_freq_your	0.2419	1.2737
word_freq_font	0.2013	1.223
word_freq_000	2.2452	9.4426
word_freq_money	0.4264	1.5317
word_freq_hp	-1.9204	0.1465
word_freq_hpl	-1.0402	0.3534
word_freq_george	-11.7673	0
word_freq_650	0.4454	1.5612
word_freq_lab	-2.4864	0.0832
word_freq_labs	-0.3299	0.719
word_freq_telnet	-0.1702	0.8435
word_freq_857	2.5488	12.7917
word_freq_data	-0.7383	0.4779
word_freq_415	0.6679	1.9501
word_freq_85	-2.0554	0.128
word_freq_technology	0.9237	2.5186
word_freq_1999	0.0465	1.0476
word_freq_parts	-0.5968	0.5506
word_freq_pm	-0.865	0.421
word_freq_direct	-0.3046	0.7374
word_freq_cs	-45.0481	0
word_freq_meeting	-2.6887	0.068
word_freq_original	-1.2471	0.2873
word_freq_project	-1.5732	0.2074
word_freq_re	-0.7923	0.4528
word_freq_edu	-1.4592	0.2324
word_freq_table	-2.3259	0.0977
word_freq_conference	-4.0156	0.018
char_freq_ ;	-1.2911	0.275
char_freq_(	-0.1881	0.8285
char_freq_[	-0.6574	0.5182
char_freq_ !	0.3472	1.4151
char_freq_$	5.336	207.683
char_freq_#	2.4032	11.0581
capital_run_length_average	0.012	1.0121
capital_run_length_longest	0.0091	1.0092
capital_run_length_total	0.0008	1.0008

TABLE 1 – Coefficients et "odd ratio" de chacune des variables

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.891     0.046      0.926     0.891     0.908      0.971    1
                 0.954     0.109      0.931     0.954     0.942      0.971    0
Weighted Avg.    0.929     0.084      0.929     0.929     0.929      0.971
```

FIGURE 1 – Performances du modèle

```
=== Confusion Matrix ===

   a   b   <-- classified as
 550  67 |   a = 1
  44 903 |   b = 0
```

FIGURE 2 – Matrice de confusion

2.3

Donnez un exemple de contre-mesure de type Tokenization attack qu'un spammeur pourrait facilement utiliser afin de contourner un filtre basé uniquement sur la fréquence d'apparition de certains mots. Votre méthode ne doit pas modifier la signification du message et ne doit pas ajouter de nouveaux mots. Appliquez votre méthode au message suivant :

DEAR RECEIVER,
You have just received a Taliban virus. Since we are not so technologically advanced in Afghanistan, this is a MANUAL virus. Please click on this link (http ://clickme.com) to delete all the files on your hard disk yourself and send this mail to everyone you know. Thank you very much for helping us.

-Taliban hacker.

D'après la documentation, "*A word [...] is any string of alphanumeric characters bounded by non-alphanumeric characters or end-of-string*". Ainsi, il suffit de couper les mots ayant un odd-ratio élevé avec un caractère non alphanumérique afin qu'ils ne soient plus pris en compte lors de la détection. On peut par exemple le faire avec un tiret et un retour à la ligne. On considère que les mots de ce mail avec un odd-ratio élevé sont *virus*, *Afghanistan*, *delete*, *everyone*, *helping* et *hacker*.

DEAR RECEIVER,
You have just received a Taliban vi-
rus. Since we are not so technologically advanced in Afgha-
nistan, this is a MANUAL vi-
rus. Please click on this link (http ://clickme.com) to de-
lete all the files on your hard disk yourself and send this mail to every-
one you know. Thank you very much for hel-
ping us.

-Taliban ha-
cker.

3. Apprentissage automatique

Une autre catégorie de méthodes qui a fait ses preuves en détection de spam consiste à s'inspirer de l'intelligence artificielle et de recourir à des techniques dites d'apprentissage automatique (machine learning). Ces méthodes ont l'avantage de s'adapter et d'apprendre pour continuellement améliorer leur performance.

3.1

Les méthodes d'apprentissage automatique peuvent être divisées en plusieurs catégories. Parmi les plus courantes, nous retrouvons l'apprentissage non-supervisé, semi-supervisé, et supervisé. Expliquez les caractéristiques de chacune de ces méthodes.

Pour l'analyse et la catégorisation de grande quantité de données, les réseaux de neurones sont de plus en plus utilisés. Pour pouvoir utiliser un réseau de neurones il faut cependant l'entrainer. Dans cette partie, nous allons présenter 3 méthodes pour entrainer un réseau de neurones.

L'apprentissage non-supervisé. Dans cette méthode, c'est l'algorithme qui effectue le tri de classe. Pour cela, il les traites comme un ensemble de variable aléatoire. L'apprentissage non-supervisé a donc l'avantage de ne pas nécessiter un expert. [3]

L'apprentissage supervisé. L'apprentissage supervisé consiste à donné un jeu de donnée étiqueté au réseau de neurones. Contrairement à l'apprentissage non-supervisé, cette méthode nécessite d'étiqueter tout le jeu de donnée par un expert. Une fois le réseau "entrainé", il devrait être capable de catégorisé une entrée automatiquement.

L'apprentissage semi-supervisé. Cette méthode regroupe l'apprentissage supervisé et non-supervisé. Dans l'apprentissage semi-supervisé, on utilise deux set de données, un étiqueté et un non étiqueté. Celui-ci permet de trier plus facilement des grands ensembles de données. [2]

a	b	classification
583	34	a=1
310	637	b=0

Table 2 – Matrice de confusion pour bayésienne

3.2

La classification naïve bayésienne (naive bayes classifier) est un exemple de méthode qui peut être utilisée afin de résoudre des problèmes de classification par apprentissage supervisé. Appliquez cette méthode au fichier spambase afin de filtrer les messages en fonction des 57 variables continues. Utilisez 66% des données pour la phase d'apprentissage.
Évaluez et discutez des performances de votre modèle en termes de : taux de vrai positif, taux de faux positif, précision, sensibilité (recall), « F-measure » et l'aire sous la courbe (ROC area). Donnez la matrice de confusion.

La classification bayésienne naïve est une méthode qui considère que toutes les classes sont indépendantes. Dans le cas de la recherche de SPAM, cela revient à dire que la probabilité d'apparition d'un mot ne dépend des autres.[5]

- **Taux de faux positif** : 0,163
- **Taux de vrai positif** : 0,78
- **Précision** : 0,832
- **Sensibilité** : 0,780
- **«F-Measure»** : 0,781
- **Aire sous la courbe** : 0,935

Comme expliqué dans la question 2.2, dans un système de filtrage de spam, nous voulons éviter au maximum les faux positif et on peut accepter quelques vrais positifs. Toutefois, on voit que dans le cas de la méthode bayésienne, nous avons plus de 16% de faux positif avec seulement 78% de message correctement détecté. Cette méthode est donc mauvaise vue que nous avons plus de 22% du spam non filtré. Nous rejetons donc plus de message légitime que de spam.
Dans la matrice de confusion est donnée dans le tableau 2. La matrice montre clairement les problèmes de classement avec beaucoup de "a" qui se retrouvent en "b".
Étant donné que la méthode bayésienne s'applique sur des données non corrélé, il est normal que les résultats ne soit pas bon. En effet lors de l'écriture d'un texte, les différents mots utilisé sont liés. Par exemple, si l'on met le mot bonjour, il y a une forte chance qu'il soit suivi d'un prénom.

3.3

Une autre méthode de classification, les forêts d'arbres décisionnels (random forest), consiste à effectuer un apprentissage sur plusieurs arbres de décisions. Appliquez cette méthode au fichier spambase et utilisez les 57 variables continues. Utilisez 66% des données pour la phase d'apprentissage. Évaluez et discutez des performances de votre modèle en termes de : taux de vrai positif, taux de faux positif, précision, sensibilité (recall), « F-measure » et l'aire sous la courbe (ROC area). Donnez la matrice de confusion.

a	b	classification
575	42	a=1
33	914	b=0

Table 3 – Matrice de confusion pour random forest

La méthode «random forest» consiste à diviser l'ensemble de donnée en de multiple sous ensemble. Cela permet donc de déterminer des probabilités relationnelles entre les classes, et donc d'avoir une meilleur répartition.[6]

- **Taux de faux positif** : 0,055
- **Taux de vrai positif** : 0,952
- **Précision** : 0,952
- **Sensibilité** : 0,952
- **«F-Measure»** : 0,952
- **Aire sous la courbe** : 0,987

Avec cet algorithme, nous avons seulement 5% de faux positif ce qui est un résultat plutôt bon. Nous avons 95% de vrai positif ce qui représente 5% de message légitime classé comme spam. Ce résultat est moins bon vue que l'on est prêt à recevoir quelques spam mais que l'on veut absolument évité de ne pas recevoir de courriel légitime.

3.4

Donnez un exemple de contre-mesure de type Statistical attack qu'un spammeur pourrait utiliser afin d'échapper à un filtre basé sur la fréquence des mots en utilisant une méthode d'apprentissage automatique. Votre exemple de contre-mesure peut impliquer de modifier le contenu du message. Proposez une solution qui permettrait de contrecarrer cette contre-mesure.

Une contre mesure qui permettrais de contrecarrer la méthode d'apprentissage automatique pourrait être l'utilisation de synonymes sur les termes important. En effet, avec la sélection aléatoire de certain mot, nous réduisons le nombre de messages qui paraissent similaires.
Pour se prémunir de ce genre d'attaque, il faudrait alors utiliser une classification des terme utilisé dans le texte. On pourrait en effet utiliser le dictionnaire de synonymes pour identifier les mots dans des catégories communes. Il faudrait pour cela ne plus considérer les mots mais simplement leur signification.

4. Performance et optimisation

4.1

Comparez et discutez, en termes de performance, les résultats que vous avez obtenus pour les différentes méthodes utilisées (régression logistique, classification naïve bayésienne, forêts d'arbres décisionnels). Selon vos résultats, quelles méthodes semblent donner les meilleures performances pour le jeu de données spambase ?

4.2

Nommez un avantage et un inconvénient d'utiliser des filtres basés sur l'apprentissage machine supervisé. Comparez avec l'apprentissage machine non supervisé en donnant un avantage et un inconvénient dans un contexte de détection de spam. Selon vous, est-ce qu'une méthode semble plus appropriée ? Justifiez votre

réponse et n'oubliez pas de donner vos références.

Les méthodes de classification supervisé ont l'inconvénient de nécessité un expert capable de définir les caractéristiques du jeu de donnée en entré.[4] Cependant, elles permettent des précisions plus élevé. Toutefois, cette méthode pose le problème d'être limité dans le temps. En effet, il faut ré-entrainer le réseau de neurones, si l'on veut ajouter la détection des nouveau spam.
Les méthodes d'apprentissage non supervisé propose des résultat de classification moins performant. Ils ont toutefois l'avantage de ne pas nécessité d'intervention humaine. Finalement, il propose une efficacité semblable entre des que l'attaque soit connu ou non connues. Le besoin de ré-entrainer le réseau de neurones est plus faible. [8]
Finalement, une bonne approche serait d'utiliser le réseau de neurones avec un apprentissage par renforcement[1], on pourrait ainsi demander à l'utilisateur de détecter les nouveaux spam.

4.3

Le jeu de données spambase est principalement basé sur la fréquence d'apparition de certains mots et sur la présence de lettres majuscule. Donnez au moins deux autres exemples de caractéristiques (features) qui pourraient être prises en compte afin d'améliorer la performance des modèles de classification que vous avez développés. Donnez vos références.

En se basant sur l'article [7] donné sur Moodle, on peut améliorer la performance de notre algorithme de classification en y intégrant les caractéristiques suivantes :

- Des informations du header comme le champs "FROM", la date, la taille du message ou encore le nombre de pièces jointes
- Les images présentes dans le mail, car du texte peut y être écrit
- L'ordre d'apparition des mots

Références

[1] Apprentissage par renforcement, Nov. 2015. Page Version ID : 120752876.

[2] Apprentissage semi-supervisé, Oct. 2015. Page Version ID : 119564802.

[3] Apprentissage non supervisé, Feb. 2016. Page Version ID : 123222753.

[4] Apprentissage supervisé, Feb. 2016. Page Version ID : 123222743.

[5] Classification naïve bayésienne, Jan. 2016. Page Version ID : 122381109.

[6] Random forest, Jan. 2016. Page Version ID : 700408350.

[7] E. Blanzieri and A. Bryl. A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review*, 29(1) :63–92, 2008.

[8] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck. Learning Intrusion Detection : Supervised or Unsupervised ? In F. Roli and S. Vitulano, editors, *Image Analysis and Processing – ICIAP 2005*, number 3617 in Lecture Notes in Computer Science, pages 50–57. Springer Berlin Heidelberg, Sept. 2005. DOI : 10.1007/11553595_6.