ivegotmail – compte-rendu – Classification Binaire Spam/NoSpam

Ben Kirane Malik, Mouhoubi Fatima

4 mars 2018

Résumé

Nous nous intéressons au probléme de classification d'une base de courriers éléctroniques (emails). Nous souhaitons à partir du corps d'un email savoir s'il s'agit d'un spam ou non. Il s'agit d'inférer cette connaissance à partir d'un corpus d'emails avec une approche basée sur les probabilités (relation de Bayes). Cette approche est présentée en première partie. L'objet de ce compte-rendu est ensuite d'étudier différentes modélisations jjavec des descriptions enrichies pour répondre au problème de classification.

Table des matières

1 Classifieur par Inférence Bayésienne

Dans cette partie, pour établir à partir des distributions de probabilités estimées dans la phase d'apprentissage si un email est un spam ou non, nous illustrerons une solution á cette problématique avec un modèle très simple. Nous travaillerons tout le long sur deux phases distinctes : la phase d'apprentissage et la phase d'évaluation ou de prédiction pour discuter de la qualité du modèle choisis.

Il est naturel de s'intéresser à la description d'un email, description que nous noterons $D(x)$ ou $\hat{x}$, s'il s'agit d'un spam nous lui associerons une étiquette de valeur $+1$, sinon de valeur -1. Au départ avec un ensemble d'apprentissage $E = \{(x, y), y \in \{-1, +1\}\}$ et formellement, nous cherchons une application $f_{\Theta(E)}$ (classifieur), telle que pour tout email x non nécessairement dans E, $f_{\Theta(E)}(x)$ soit la classe de x.

La phase d'apprentissage est donc la phase où on estimera le modèle $\Theta(E)$ par inférence sur E et la phase d'évaluation est la phase où on calcul l'image de $f_{\Theta(E)}$ pour un ensemble d'emails.

Nous considérons deux variables aléatoires :

1. X prendra les valeurs possibles des descriptions pour un email
2. Y vaut $+1$ ou bien -1 selon qu'il s'agit respectivement d'un spam ou non.

Notre approche est probabiliste : on souhaite par intuition connaître les distributions

$$P(Y = y|X = \hat{x}), \quad y \in \{-1, +1\} \tag{1}$$

et pouvoir les comparer pour espérer ensuite prédire si $D^{-1}(\hat{x})$ (l'email décrit par $\hat{x}$) est un spam ou encore si l'application $\hat{f}_E \colon Im(X) \to \{+1, -1\}$ apprise avec la description D indique pour un email x sa classe simplement en calculculant $\hat{f}_E[D(x)]$.

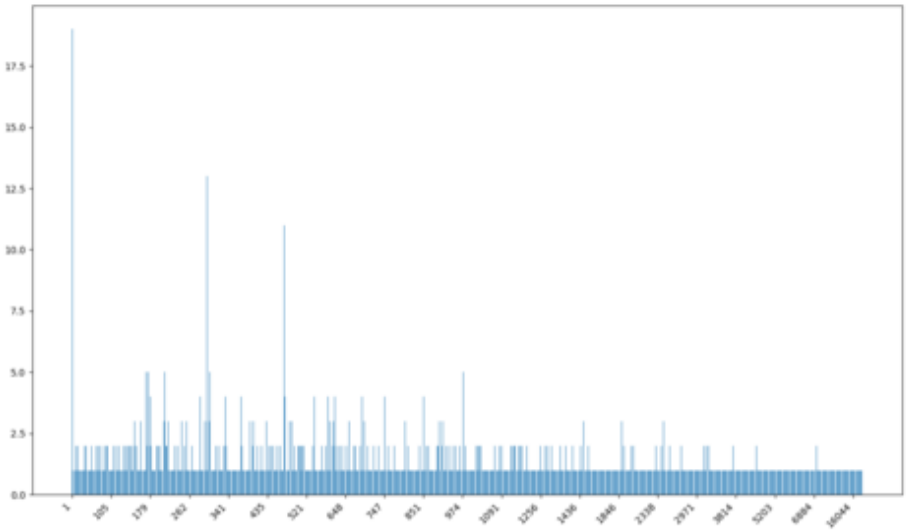
La frontière de décision peut se calculer en utilisant la relation de Bayes sur les probabilités conditionnelles (1)

$$\frac{P(X = \hat{x}|Y = +1)P(Y = +1)}{P(X = D(x))} - \frac{P(X = \hat{x}|Y = -1)P(Y = -1)}{P(X = D(x))} = 0 \tag{2}$$

(2) est l'estimation qu'un email décrit par $\hat{x}$ est un spam (resp. n'est pas un spam) pondérée par le ratio entre le nombre de spam (resp. non spam) et des emails dont les descriptions sont identiques.

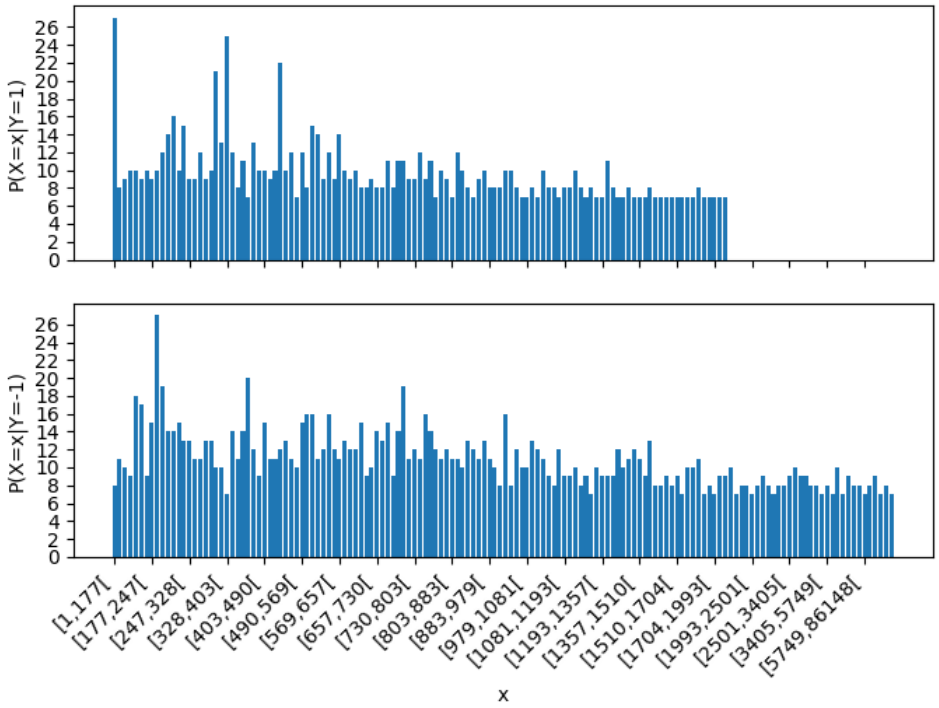
Nous considérons la description $D\colon x \mapsto l(x)$ tel que $l(x) \in \mathbb{N}$ est la longueur du corps d'un email. Nous estimons, pour cette description, étant donné une classe sur l'ensemble d'apprentissage, en comptant le nombre d'emails pour chaque description apparue lors du parcours de l'ensemble d'apprentissage. Il s'agit d'utiliser l'estimateur de fréquence pour la probabilité $P(X = D(x)|Y = y)$. Puis nous estimons la distribution de $P(Y = y)$ parce que l'ensemble d'apprentissage est fini. La distribution sur la classe des spams de notre estimateur est représentée sur la FIGURE 1. On lit une valeur prise par $l(x)$ en abscisse et l'effectif compté en ordonnée.

FIGURE 1 – Estimation sur la longueur des spam



Plutôt que garder toutes les valeurs apparues pour $l(x)$ nous les regroupons dans des intervalles $[i_k, s_k]$ telles que pour tout intervalle décrit par son indice k, $\text{Card}\left(\{\ x \mid l(x) \in [i_k, s_k\}\right) = 5$ par exemple (voir FIGURE 2).

FIGURE 2 – Estimation regroupée des distributions pour la description par longueur



Nous avons donc validé la phase d'apprentissage sur le corpus E et obtenu le modèle $\Theta(E)$ défini par deux fonctions : $e\colon \mathbb{N} \to [0,1]$ l'estimation de probabilité pour les email x décrits pour $l(x)$ sachant que x est un spam et $\overline{e}\colon \mathbb{N} \mapsto [0,1]$ l'estimateur sachant qu'il ne s'agit pas d'un spam ; et par $p = \text{Card}\left(\{(x,y) \in E \mid y = 1\}\right)/\text{Card}(E)$ (et $\overline{p} = 1 - p$).

Plus précisément, les fonctions e et $\overline{e}$ sont définies sur une partition finie de $\mathbb{N}$ comme indiqué pour construire les intervalles $[i_k, s_k]$ dont il est question plus haut. La prédiction s'évalue avec le signe de l'expression

$$e(\hat{x}) \cdot p - \overline{e}(\hat{x}) \cdot (1 - p) \tag{3}$$

avec les notations décrites. Donc en pratique on retournera l'estimation sur l'intervalle appris dans lequel $\hat{x} = l(x)$ se trouve. C'est justement ce qui a été implémenté pour ce cas de figure oú le descripteur D est une fonction dont les images sont finies et sont des parties $\mathbb{N}$.

Pour finir d'illustrer avec l'étude de ce premier classifieur, nous souhaitons valider ou infirmer ses valeurs de prédictions. Pour ce faire nous divisons l'ensemble E d'apprentissage en deux sous-ensembles A et T. L'ensemble A pour la phase d'apprentissage et l'ensemble T pour estimer l'erreur des prédictions. Pour cette démonstration, nous choisissons arbitrairement 80% des emails étiquetés dans E pour constituer A et les emails restants constituent T. Ce choix arbitraire est alors réalisé 40 fois et à chaque découpage nous notons le nombre de prédictions incorrectes. La moyenne de la probabilité d'erreur et le score du meilleur classifieur sont présentés dans la TABLE 1.

TABLE 1 – Estimation de l'erreur

	$\min_f(P(f(x) \neq y))$	en moyenne
sur T	0.5454124189063948	0.5641682113067655
sur A	0.48148148148148145	0.515787037037037

Autant dire que les modèles appris avec une telle application de description ne sont pas fiables. Cependant, ce modèle nous a permis de détailler chaque phase dans la résolution du problème de classification posé. Nous allons nous porter plus longtemps à présent sur le choix de la description D pour compléter le manque d'informations qui nous empêche d'inférer correctement la classe à laquelle appartient un email.

2 Description vectorielle d'emails, représentations, cas limites

L'objet d'étude, ici, est de réfléchir et d'évaluer la performance de modèles appris sur des descriptions enrichies.

Avec la description que nous avions choisi pour l'exemple de la première partie nous avons omis jusque là le corps de l'email. Il pourrait s'agir sur papier de décomposer le corps en mots et de représenter un email $x = \left(x_{i_1}, x_{i_2}, \ldots, x_{i_{l(x)}}\right)$ avec $\mathcal{D} = \{x_1, x_2, \ldots, x_d\}$ le dictionnaire des d mots possiblement trouvables dans un email. Nous pouvons imaginer plusieurs descriptions pour cette représentation de x :

$$
\begin{aligned}
D_0(x) &= (\mathbb{I}_x[x_i])_{i\in\{1,\cdots,d\}} \\
D_1(x) &= (\mathbb{I}_x[x_i, x_j])_{i\neq j\in\{1,\cdots,d\}} \\
&\cdots \\
D_n(x) &= (\mathbb{I}_x[x_{i_1}, x_{i_2}, \ldots . x_{i_n}])_{i_1\neq i_2\neq\cdots\neq i_n\in\{1,\cdots,d\}}
\end{aligned}
$$

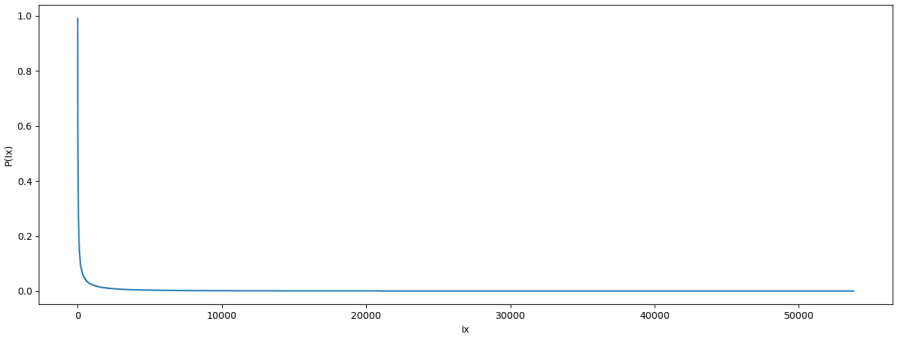
où $\mathbb{I}_x[x_{i_1}, x_{i_2}, x_{i_n}]$ vaut 1 si x contient la sous-séquence de mots $(x_{i_1}, x_{i_2}, \ldots, x_{i_n})$ et 0 sinon.

Par intuition, D_0 est une description moins riche que D_1, ..., D_n, avec pour hypothèse très raisonnable que $n << d$. Mais il en est pas des moindres d'envisager de les implémenter. En effet la complexité pour représenter l'estimation $P(X = D_0(x)|Y)$ en mémoire est en $O(2^d)$ puisque nous représenterons les valeurs de probabilités pour les éventuelles 2^d descriptions dans $\{0,1\}^d$. Sur les corpus `spam`, `nospam` fournis, on dénombre 53808 mots. C'est à dire qu'il serait nécessaire d'avoir une structure de données donnant l'accès à 2^{53808} , et autrement dit environs $10^{1800\cdot 9}$ flottants.

Face à ce problème nous pouvons ou bien étudier l'indépendance des variables aléatoires $X_i = \mathbb{I}_x[x_i]$ et en conservant la description D_0 stocker l'estimation $P(X = D_0(x)|Y)$ avec $O(d)$ flottants $P(X_i = \mathbb{I}_x[x_i]|Y)$, ou bien réduire d à partir d'une heuristique parce que la borne 2^d reste très pessimiste.

Faute de quoi, on peut s'apercevoir qu'il n'est pas nécessaire de considérer $\mathcal{D}$ dans sa totalitée aussi bien qu'il n'est pas nécessaire de sauvegarder toutes les combinaisons possibles d'apparitions parce que nombre d'entre elles sont nulles i.e. impossibles de la perspective de l'ensemble d'apprentissage.

FIGURE 3 – Fréquences des mots dans le corpus



La FIGURE 2 propose une première heuristique qui est de ne pas considérer les mot trop peu apparus avec chacune des indicatrices au delà des 5000 premiers mots de $\mathcal{D}$ et de ne pas considérer les mots trop frèquents apparus dans l'ensemble d'apprentissage non plus. Dès à présent nous concentrerons nos efforts sur $D_{\sigma,\overline{\sigma}}$ la description pour un email sur le sous-ensemble significatif de $\mathcal{D}$, σ, pour lequel nous pourrions estimer non exhaustivement et raisonnablement les combinaisons apparues. Donc il s'agit de la restriction de D_0 sur σ ; et d'autre part prolongée sur le sous-ensemble non significatif $\overline{\sigma}$ sur lequel on présentera plus loin des alternatives sous des hypothèse faibles d'indépendance. On nomme se couple $(\sigma, \overline{\sigma})$ la représentation du corpus.

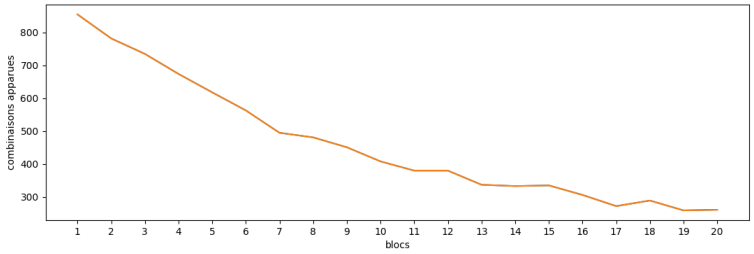
Quand il s'agira de proposer une description dans ce contexte, nous distinguerons
— l'estimation basée sur le descripteur D_σ avec, par exemple σ les 500 mots les plus fréquents
— l'estimation basée sur les 500 premiers descripteurs $D_{i,\sigma}$ pour chacun de ces 500 mots de σ.

D'ailleurs nous nous proposons de visualiser empiriquement si le produit des distributions indépen-

dantes des $P(X_i = D_{i,\sigma}(x)|Y)$ est proche de la distribution de $P(X = D_\sigma(x)|Y)$ pour le corpus donné. Nous réalisons autant d'estimations qu'il y a de blocs successifs de 500 mots dans $\mathcal{D}$ tronqué au mot 10000. Il s'agit de 20 blocs, la complexité empirique du nombre d'estimations pour chaque bloc se lit TABLE 2). Nous prenons les mots dans l'ordre des plus fréquents aux moins fréquents.

TABLE 2 – Évaluation empirique de la complexité de représentation

```
[0:10] 855, 782, 735, 674, 618, 563, 495, 481, 451, 408
[10:20] 380, 380, 337, 333, 335, 306, 272, 289, 259, 261
```

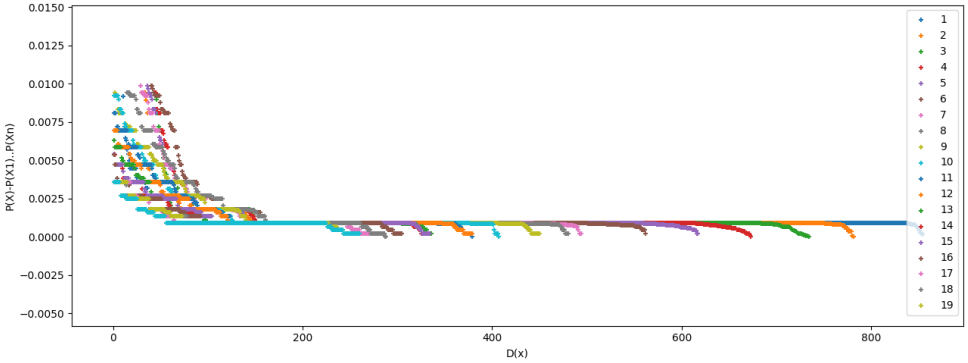


Le premier constat empirique est que la complexité de D_σ à été surestimée par 2^d dans ce cas pratique avec $\text{Card}(\sigma) = 500$. De plus, on visualise ici que l'information inférée sur le corpus est moindre sur les tranches de mots moins fréquents. Pour la représentation sur les 10000 premiers mots de $\mathcal{D}$, nous pourrions plutôt que considérer les estimations avec D_σ considérer les 20 estimations sur ces blocs avec D_{σ_i} où σ_i correspond à un bloc et supposer ces distributions indépendantes. Cette dernière hypothèse est moins forte que si on suppose l'indépendance sur des blocs de taille 1.

Le calcul des distributions pour chaque représentation σ_i, $1 \leq i \leq 20$, pour les 20 blocs de 500 mots et le calcul des distributions pour chaque représentation en supposant l'indépendance (donc 20 vecteurs de taille 500) sont réalisés séparéments en exploitant les features de la librairie `multiprocessing` de `python3` et en utilisant le module `numpy`. Nous sauvegardons ces données sur le corpus dans des fichiers qui finalement, rappelons-le, décrivent l'ensemble des paramètres pour construire le modèle $\Theta(E)$.

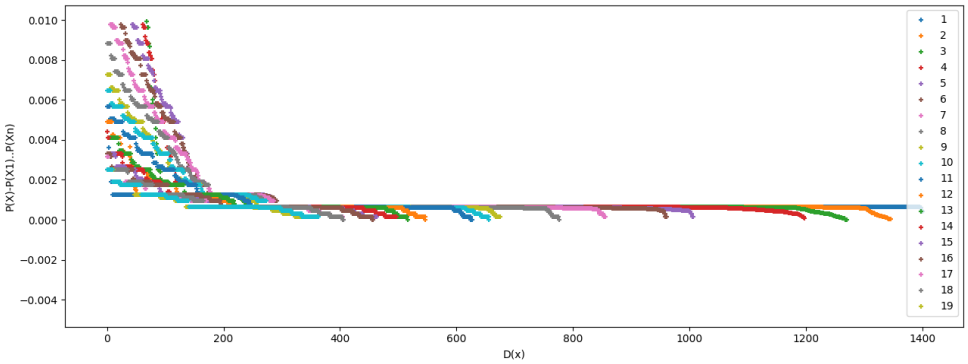
En termes d'implémentation, l'estimation d'une représentation σ_i *supposée indépendante* est une liste de taille $\text{Card}(\sigma_i)$ des probabilités $P(X = D_{j,\sigma_i}(x)|Y = y)$, $1 \leq j \leq \text{Card}(\sigma_i)$, pour chaque label $y \in \{-1, +1\}$ et estimée sur les x de E. D'autre part, l'estimation d'une représentation σ_i est un dictionnaire dont les clés sont les chaînes de caractères de tailles $\text{Card}(\sigma_i)$ composées de `0` et `1` qui pointent vers les probabilités $P(X = D_{\sigma_i}(x)|Y = y)$ – ces choix d'implémentation se justifient du fait que $D_{\sigma_i}(x) \in \{0,1\}^{\text{Card}(\sigma_i)}$ et que $D_{j,\sigma_i}(x) \in \{0,1\}$.

FIGURE 4 – $\Delta_{+1}(x) = \left|\prod_j P(X = D_{j,\sigma_i}|Y = +1) - P(X = D_{\sigma_i}|Y = +1)\right|(x)$



Nous avons calculé pour chaque bloc la différence de probabilité sous chacune des hypothèses. Les différences sont tracées sur la FIGURE 4 et la FIGURE 5 si elle est inférieure à 10^{-2} pour les estimations des représentations σ_i, $1 \leq i \leq 19$. On en déduit que l'hypothèse d'indépendance des $X_j = D_{j,\sigma_i}$ est

FIGURE 5 – $\Delta_{-1}(x) = \left|\prod_j P(X = D_{j,\sigma_i}|Y = -1) - P(X = D_{\sigma_i}|Y = -1)\right|(x)$



trop forte pour inférer un bon modèle et surtout sur les blocs des mots les plus fréquents. On pourra se faire une idée plus fine des indépendances en fonction de σ_i le bloc de mots évalué avec la FIGURE 6 et la FIGURE 7 qui pour chaque classe et chaque bloc indiquent le nombre de clés pour lesquelles ont doit réfuter l'hypothèse pour un seuil $\Delta_y(x) < 10^{-2}$.

FIGURE 6 – Seuils d'indépendances pour le corpus spam

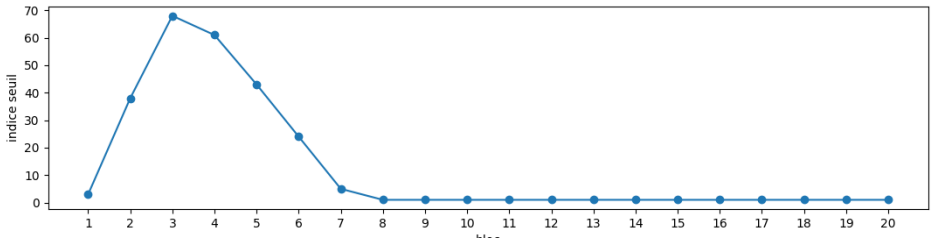
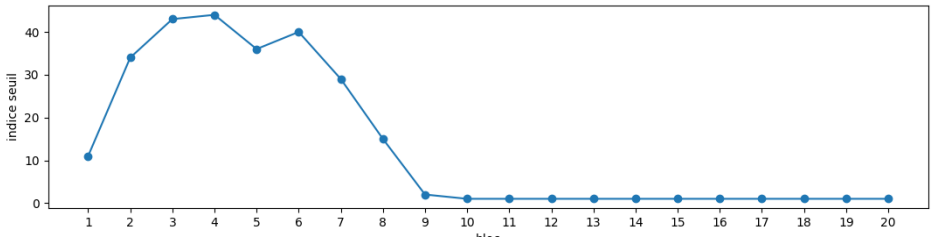


FIGURE 7 – Seuils d'indépendances pour le corpus nospam



Cette analyse nous conduit à ne surtout pas faire l'hypothèse d'indépendance sur les premiers descripteurs correspondants aux mots les plus fréquents. Aussi plus les probabilités sont faibles, plus il est possible que les valeurs transfèrent une erreur d'estimation suite à plusieurs opérations numériques. À ce sujet, nous avons pris quelques précautions comme faire priorité aux additions et soustractions face aux multiplications et aux divisions et donc privilégié le passage au logarithme comme avec (4) ce qui permet de réduire les erreurs.

$$\log\left(\prod_i p_i\right) = \sum_i \log(p_i) \tag{4}$$

À présent, nous avons posé les notations qui nous permettent de décrire les différentes représentations envisageables pour construire un modèle d'inférence sur un ensemble d'apprentisage ; et nous avons visité sur un corpus d'exemples d'apprentissages les écueils qui pourraient nuire à la qualité du modèle ou le rendre discutable pour sa complexité. Cependant, et nous l'avons observé plusieurs fois, restreindre la

description à un sous-ensemble σ de mots n'est pas satisfaisant parceque l'on perds une quantité non négligeable d'information.

3 Enrichissement au moindre coût : notre modèle

Après avoir parcouru le registre des descripteurs, dans cette partie nous présentons le classifieur contruit sur une description étendue avec une représentation $(\sigma(E), \overline{\sigma}(E))$ évaluée selon E au début de l'apprentissage.

Nous présenterons d'abords les critères et la procèdure pour constuire cette représentation. C'est le couple de deux ensembles : un ensemble de mots et un ensemble des parties des mots laissés de côté ; qui nous permettra par la suite de constuire un modèle $\Theta(E)$ avec une estimation d'erreur faible.

Nous devrons à mi-temps présenter un dernier descripteur $D_{\overline{\sigma}}$.

3.1 Constuire une représentation $\sigma, \overline{\sigma}$ satisfaisante

Reprenons à la FIGURE 2 : c'est l'allure de la courbe des fréquences d'apparition des mots dans le corpus qui nous fait induire que les mots trop fréquemment présents augmentent non significativement la complexité pour représenter le modèle d'autant que ces mots sont très succeptibles d'augmenter le nombre de combinaisons $D_\sigma(x)$ quand on visite E pour apprendre le modèle. D'autre part, nous avons vu qu'au delà des 10000 premiers mots les plus fréquents, les mots sont présent très spécifiquement pour peu d'exemples du corpus.

La procédure pour constituer la représentation $(\sigma, \overline{\sigma})$ qui nous intéresse ce décompose en trois étapes

1. vider $\mathcal{D}$ des mots facteurs de bruit parce qu'ils apparaissent trop fréquement,
   on notera $\mathcal{D}^0$ le nouveau dictionnaire
2. déterminer l'ensemble σ
   en recoupant avec un nombre raisonable des mots les plus fréquents de $\mathcal{D}^0$,
   on notera $\mathcal{D}^\sigma$ le dictionnaire obtenu et
   il reste $\mathcal{D}^1$ le dictionnaire des mots les moins significativement importants
3. choisir une partition de $\mathcal{D}^1$ satisfaisante dont on déduit déduire $\overline{\sigma}$

Supposons que nous avons déjà déterminer σ et que nous arrivons à la dernière étape avec $\mathcal{D}^1 = \{x_1, \ldots, x_n\}$ tel que $f(x_1) \geq \ldots \geq f(x_n)$. Alors n est certainement encore très grand. Nous aurions intêret à résumer l'information sur des parties succesives de $\mathcal{D}^1$. Pour déterminer les tranches, nous posons la suite des $n-1$ él'ements $d_i \colon i \mapsto f(x_{i+1}) - f(x_i)$. Les valeurs qui nous interessent sont les valeurs de i telles que $d_i < 0$. Alors que n est grand, $f(x_1)$ est petit et de plus en plus communément, les x_i sont de même fréquence et donc $d_i = 0$ pour ces valeurs. Si on note $(i_1, i_2, \ldots i_k)$ les valeurs de i pour lesquelles $d_i < 0$ on obtient les indices qui délimitent les tranches de mots donnant en moyenne la même quantité d'information.

Avec les notations utilisées, la représentation

$$\overline{\sigma} = \left\{ \{x_{i_1}, \ldots, x_{i_2}\}, \{x_{i_2+1}, \ldots, x_{i_3}\}, \ldots, \left\{x_{i_{k-1}+1}, \ldots, x_{i_k}\right\} \right\} \tag{5}$$

suggère une description moins lourde qui apporte une information supplèmentaire pour le modèle. Nous choisissons

$$D_{j,\overline{\sigma}} \colon x \mapsto \sum_{i=i_j}^{i_{j+1}} \mathbb{I}_x[x_i] \tag{6}$$

les k descripteurs à valeurs dans $\mathbb{N}$ comptant le nombre de mots apparues dans un email x dans la tranche $[i_j, i_{j+1}[$ de $\overline{\sigma}$. Avec le même corpus que dans les parties précédentes nous convenons par tatonnement que si la dérivée discrète de la fréquence est strictement négative plus de 500 fois sur un intervalle alors on peut le considérer comme un partie de $\overline{\sigma}$.

On utilise un critière similaire pour déterminer l'ensemble des mots significativement présents dans $\mathcal{D}^0$: on considère les mots dont les fréquences ordonnées du plus au moins fréquents est au moins strictement décroissante sur des intervalle de 10 mots. Ainsi on se limitera à une représentation σ de taille 642 mots ce qui est satisfaisant d'après la partie 2.

Finalement nous avons retranché arbitrairement à la première étape les mots trop fréquents à l'étape 1 en selectionnant les mots dont la fréquence est inférieure à $\frac{1}{2}\cdot\text{Card}(\{Y=1\})/\text{Card}(E)$.

3.2 Notre modèle : 0.05 prédictions éronnées en moyenne

Exactement à l'image de la première partie l'ensemble d'apprentissage est divisé aléatoirement et arbitrairement en deux sous-ensemble de E : les exemples de test T (30%) et les exemples d'apprentissage A (70%).

Après avoir évalué la représentation $(\sigma, \overline{\sigma})$ sur A, l'apprentissage se décompose en deux parties indépendantes : le calcul des estimations pour la description σ et le calcul des estimations pour les descriptions de $\overline{\sigma}$.

En moyenne, l'estimation des probabilités $P(D_\sigma|Y)$ sur les corpus `spam` et `nospam` (2698 emails) s'opère sur une représentation de 500 à 680 mots relativement rapidement. D'autre part, pour chaque partie $\overline{\sigma}_i$ de $\overline{\sigma}$ nous somme ramenés au cas de la première partie parce que les descripteurs $D_{i,\overline{\sigma}}$ correspondants sont des applications à valeurs dans $\mathbb{N}$. Alors, pour s'assurer de pouvoir estimer la probabilité correspondante et représenter en mémoire cette distribution, nous devons calculer un regroupement des valeur prisent par les descriptions. Cela dit, le modèle apprend en un temps raisonable malgrès que cette second partie est relativement plus coûteuse en temps de calul.

Les résultats pour les calculs de ces deux représentations sont supposés indépendants puisque lorsqu'il s'agit de calculer $P(X=x|Y=y)$ on évalue

$$P(X = D_\sigma(x)|Y=y)\prod_i P(X = D_{i,\overline{\sigma}}(x)|Y=y). \tag{7}$$

En exploitant les features des outils `multiprocessing` et `multiprocessing.dummy` de `python3`, le modèle $\Theta(A)$ est appris et testé sur T globalement en 4 minutes (Table 3).

Table 3 – Temps pour apprendre et tester le modèle

real	0m40.460s
user	3m26.552s
sys	0m6.872s

Nous avons entraîné 11 modèles avec, à chaque fois, un découpage différent de E. En moyenne un modèle prédit correctement 94.29% des emails. Le meilleur modèle entraîné sur cette représentation du corpus se trompe dans 5.19% des cas.

Conclusion – Nous avons du trouver un compromis pour représenter un email au risque de ne pas pouvoir calculer les distributions de probabilités nécessaires pour prédire au mieux si un nouvel email est un spam ou ne l'est pas. Ce compromis est réalisé parce que nous avons fait les choix de cette représentation. C'est sur ces choix que nous devrions réfléchir si nous souhaiterions améliorer les modèles ou réaliser qu'ils sont suffisament satisfaisant à l'égard de cette approche.