UNE NOUVELLE APPROCHE POUR LA DÉTECTION DES SPAMS SE BASANT SUR UN TRAITEMENT DE DONNÉES CATÉGORIELLES

par

Yassine Z. Parakh Ousman

Mémoire présenté au Département d'informatique
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES

UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, 28 juin 2012

Your file Votre référence
ISBN: 978-0-494-88906-0

Our file Notre référence
ISBN: 978-0-494-88906-0

Le 5 juillet 2012

le jury a accepté le mémoire de Monsieur Yassine Zaralahy Parakh Ousman dans sa version finale.

Membres du jury

Professeur Shengrui Wang
Directeur de recherche
Département d'informatique

Professeur André Mayers
Membre
Département d'informatique

Professeure Hélène Pigot
Présidente rapporteur
Département d'informatique

Sommaire

Le problème des spams connaît depuis ces 20 dernières années un essor considérable. En effet, le pollupostage pourrait représenter plus de 72% de l'ensemble du trafic de courrier électronique. Au-delà de l'aspect intrusif des spams, ceux-ci peuvent comporter des virus ou des scripts néfastes ; d'où l'intérêt de les détecter afin de les supprimer. Le coût d'un envoi de courriels par un spammeur étant infime, ce dernier peut se permettre de transmettre le spam au plus d'adresse de messagerie électronique. Pour le spammeur qui arrive à récupérer même une petite partie d'utilisateurs, son opération devient commercialement viable. Imaginant un million de courriels envoyés et seul 0,1% de personnes qui se font appâtées, cela représente tout de même 1 millier de personnes ; et ce chiffre est très réaliste. Nous voyons que derrière la protection de la vie privée et le maintien d'un environnement de travail sain se cachent également des enjeux économiques.

La détection des spams est une course constante entre la mise en place de nouvelles techniques de classification du courriel et le contournement de celles-ci par les spammeurs. Jusqu'alors, ces derniers avaient une avance dans cette lutte. Cette tendance s'est inversée avec l'apparition de techniques basées sur le filtrage du contenu. Ces filtres pour la plupart sont basés sur un classificateur bayésien naïf. Nous présentons dans ce mémoire une approche nouvelle de cette classification en utilisant une méthode basée sur le traitement de données catégorielles. Cette méthode utilise les N-grams pour identifier les motifs significatifs afin de limiter l'impact du morphisme des courriers indésirables.

Mots-clés : *spam, class, text-mining, bayésien, smtp, catégorielles, n-grams, courriel*

Sommaire

Remerciements

Je voudrais, avant tout, remercier mon directeur le professeur Shengrui Wang, qui tout au long de ma maîtrise, a su m'apporter le soutien et la motivation nécessaire pour l'accomplissement de mes travaux de recherche. Je lui suis également reconnaissant pour sa patience à mon égard, sa disponibilité pour m'expliquer et m'introduire des approches et concepts pointus. Je le remercie également pour m'avoir permis de réaliser mon projet dans des conditions favorables autant sur le plan matériel que sur l'environnement de travail.

Je souhaite remercier particulièrement mes collègues le Dr Abdelalli Kelil et Alexei Nordell-Markovits pour m'avoir aidé et avoir participé à mes travaux de recherche présentés dans ce mémoire.

Je remercie également les membres du jury les professeurs André Mayers et Hélène Pigot qui ont accepté de prendre le temps pour corriger mon travail.

Enfin, j'adresse une pensée affective à mes parents, ma famille et mes amis pour leur support tout au long de mon entreprise.

Abréviations

ASCII American Standard Code for Information Interchange

DKIM DomainKeys Identified Mail

DNS Domain Name Service

DNSBL DNS-based Blackhole List

FAI Fournisseur d'accès Internet

HTML HyperText Markup Language

IP Internet Protocol

MLE Maximum Likelihood Estimate

MTA Mail Transfert Agent

OSI Open Systems Interconnection

RBL Realtime Blackhole List

SMTP Simple Mail Transfert Protocol

SPBH Sparse Binary Polynomial Hashing

SPF Sender Policy Framework

SVM Support Vector Machine

Table des matières

Liste des figures

Liste des tableaux

Introduction

Depuis la démocratisation de l'Internet, le nombre de boîtes de courriers électroniques ne cesse d'augmenter. En effet, on peut faire le simple constat qu'aujourd'hui pratiquement tous les usagers d'Internet ont plus d'une boîte courriel. Ainsi, une communication de masse à travers ce média peut toucher un grand nombre de personnes d'autant plus que le coût d'envoi d'un courriel est dérisoire en comparaison avec le coût de l'envoi d'un courrier postal classique.

Nous recevons dans nos diverses boîtes de courriels des centaines de courriers indésirables appelés polluriels, pourriels ou encore spams. Mais qu'est-ce que réellement un spam ? Il existe en effet plusieurs définitions du spam mais toutes s'accordent sur le fait qu'il s'agisse d'une communication électronique non sollicitée et indésirable. Ce courrier peut être un simple message publicitaire ou une tentative d'hameçonnage qui est transmis à grande échelle. La plupart du temps, ces communications proviennent d'un destinataire inconnu, mais elles peuvent également être issues d'un correspondant connu dont l'identité a été usurpée. Cet envoi à grande échelle a des conséquences pour le fournisseur d'accès internet (FAI) puisque le coût du transfert et du stockage n'est pas négligeable pour lui puisqu'il va y avoir des milliers de messages qui vont transiter. Mais aussi pour l'utilisateur puisqu'il peut être noyé par ces courriers indésirables et perdre ainsi des messages importants, il peut être victime d'une usurpation d'identité ou même être lui-même victime d'une fraude. Ce phénomène est d'autant plus alarmant que selon les études de Zdziarski [48] l'envoi de spam représente entre 35% et 65% du trafic de courriels sur Internet. Il estime également la croissance de ce type de communication d'un taux annuel de 15% à 20%.

Aujourd'hui, il existe diverses approches pour détecter ces communications dites indésirables. Elles peuvent être aussi bien basées sur des techniques exploitant la particularité des protocoles de communication et d'authentification des clients que sur le filtrage du contenu. Ces deux approches sont combinées pour réduire au maximum la quantité de spams reçus. Effectivement, les logiciels anti-spam couplent plusieurs techniques et filtres pour lutter contre les spams.

Les techniques se basant sur les particularités des protocoles de communication ne peuvent englober tous les aspects du problème et font souvent face aux manquements du système. Les techniques se basant sur le filtrage du contenu quant à elles, relèvent d'un problème de classification classique en deux classes distinctes d'une part les courriers indésirables et d'autre part les courriers légitimes. Les méthodes de classification pour le filtrage sont souvent une opération statistique qui consiste à regrouper des objets (ici, les courriels) en un nombre limité de groupes (ici, deux groupes). Il s'agit d'identifier l'appartenance de nouveaux objets à un des sous-ensembles dont les caractéristiques distinctives sont connues grâce aux données d'entraînement du modèle. Ainsi, ces approches se construisent en deux phases : une première phase d'apprentissage dont la justesse va influer sur la seconde phase de classification à proprement dit. Cette approche tire ses sources des anciens filtres basés sur une liste noire de mots. Un courriel qui va contenir les mots présents sur la liste sera susceptible d'être du spam. Cette dernière montre très vite ses limites.

Le défi est de détecter le spam rapidement (avec un apprentissage court ou quand les spammeurs changent certains caractères de mots clés) et d'éviter de classer des courriers légitimes en tant que spam. Dans ce mémoire, nous allons tout d'abord explorer les principales techniques basées sur les particularités des protocoles qui sont aujourd'hui utilisées pour lutter contre le spam. Nous allons ainsi étudier le principe de leur fonctionnement pour connaître leurs avantages et désavantages. La deuxième partie de notre étude va porter sur le filtrage du contenu à l'aide d'un algorithme bayésien. Nous allons dans cette partie définir les concepts d'apprentissage et de statistique bayésienne pour comprendre le fonctionnement de tels algorithmes. Enfin, dans la troisième partie de notre étude, nous expliquerons une nouvelle approche,

CLASS [16], pour réaliser l'analyse de contenu. Nous présenterons les résultats de cette approche pour illustrer son efficacité et voir comment elle réagit par rapport à l'apprentissage. Ce dernier va nous permettre d'évaluer l'effet des changements de mots que les spammeurs utilisent, sur notre classificateur.

Chapitre 1

La lutte contre le spam

Il existe deux manières de détecter et bloquer les pourriels ; une par détection basée sur des techniques de blocage d'adresses IP et vérification d'identité ainsi qu'une autre basée sur l'analyse du contenu du courriel que nous traiterons plus tard. Ce chapitre présente une revue des outils et techniques utilisés pour bloquer les courriers indésirables.

1.1 Le tout premier spam

Peu importe la manière dont on définit le spam (courrier indésirable, polluriel ou pourriel) aujourd'hui, on s'accorde que le tout premier spam distribué sur un large réseau date de 1978. C'était une publicité d'une entreprise d'équipement numérique. Cette publicité fut diffusée sur le réseau Arpanet [1]. Les clients de messagerie n'étaient pas aussi évolués que ceux d'aujourd'hui. De ce fait, les polluposteurs (spammeurs) devaient entrer toutes les adresses des destinataires individuellement. De plus, faute d'espace tampon du programme SNDMSG qu'ils utilisaient, seulement 320 destinataires l'avaient reçu lors de la première diffusion. Voici ce fameux spam dont on a

1. ARPANET ou Arpanet (acronyme anglais de "Advanced Research Projects Agency Network", souvent typographié "ARPAnet") est le premier réseau à transfert de paquets développé aux états-Unis par la DARPA. Le projet fut lancé en 1969 et la première démonstration officielle date d'octobre 1972.

retiré les 9 pages d'adresses de destinataire [48].

> Mail-from : DEC-MARLBORO rcvd at 3-May-78 0955-PDT
> Date : 1 May 1978 1233-EDT
> From : THUERK at DEC-MARLBORO
> Subject : ADRIAN@SRI-KL
>
> DIGITAL WILL BE GIVING A PRODUCT PRESENTATION OF THE NEWEST MEMBERS OF THE DECSYSTEM-20 FAMILY; THE DECSYSTEM-2020, 2020T, 2060, AND 2060T. THE DECSYSTEM-20 FAMILY OF COMPUTERS HAS EVOLVED FROM THE TENEX OPERATING SYSTEM AND THE DECSYSTEM-10 <PDP-10> COMPUTER ARCHITECTURE. BOTH THE DECSYSTEM-2060T AND 2020T OFFER FULL ARPANET SUPPORT UNDER THE TOPS-20 OPERATING SYSTEM. THE DECSYSTEM-2060 IS AN UPWARD EXTENSION OF THE CURRENT DECSYSTEM 2040 AND 2050 FAMILY. THE DECSYSTEM-2020 IS A NEW LOW END MEMBER OF THE DECSYSTEM- 20 FAMILY AND FULLY SOFTWARE COMPATIBLE WITH ALL OF THE OTHER DECSYSTEM-20 MODELS.
>
> WE INVITE YOU TO COME SEE THE 2020 AND HEAR ABOUT THE DECSYSTEM-20 FAMILY AT THE TWO PRODUCT PRESENTATIONS WE WILL BE GIVING IN CALIFORNIA THIS MONTH. THE LOCATIONS WILL BE :
>
> TUESDAY, MAY 9, 1978 - 2 PM
> HYATT HOUSE (NEAR THE L.A. AIRPORT)
> LOS ANGELES, CA
>
> THURSDAY, MAY 11, 1978 - 2 PM
> DUNFEY'S ROYAL COACH
> SAN MATEO, CA
> (4 MILES SOUTH OF S.F. AIRPORT AT BAYSHORE, RT 101 AND RT 92)
>
> A 2020 WILL BE THERE FOR YOU TO VIEW. ALSO TERMINALS ON-LINE TO OTHER DECSYSTEM-20 SYSTEMS THROUGH THE ARPANET. IF YOU ARE UNABLE TO ATTEND, PLEASE FEEL FREE TO CONTACT THE NEAREST DEC OFFICE FOR MORE INFORMATION ABOUT THE EXCITING DECSYSTEM-20 FAMILY

L'émetteur de ce message se nommait Gary Thuerk. Il espérait que les destinataires allaient répondre à l'invitation pour apprendre sur le service de soutien ARPANET de l'entreprise. Et il a obtenu des réponses puisqu'une série de discussions controversées commença à propos de ce message et le résultat fut tel que le message avait été diffusé d'autant plus pour que tout le monde puisse réagir à ce sujet. Le fait que tout le monde ait contribué à cette controverse sur le premier spam l'a rendu d'autant plus populaire. Le spam a créé une quantité considérable de chargements sur ce que l'on considère aujourd'hui comme connexion à faible bande passante. Avec des ressources en mémoire et bande passante réduites, cette distribution peut être

considérée à la limite comme une attaque par déni de service (DoS). À la suite de cette controverse, l'entreprise a décidé de retirer la campagne et de ne plus envoyer de tel message.

1.2 Les différentes techniques basées sur le bloquage d'adresse IP utilisées dans la lutte contre le spam

Les techniques utilisées pour lutter contre le spam sont de nature variées, nous les classons de la manière suivante :

- L'identification de l'émetteur du message grâce à sa signature numérique (S/-MIME) soit par son authentification au serveur SMTP [2] (SMTP\AUTH).
- Les protocoles visant à vérifier la provenance du courriel (SPF) ou leur authenticité (DKIM, Domain Keys, IIM, etc.).
- L'analyse comportementale qui permet d'identifier le spam en exploitant leurs comportements caractéristiques : non-respect des RFCs [3], cadence des messages reçus, etc.
- Le filtrage par signature en utilisant les bases de données connues des terminaux qui transmettent des courriers indésirables (RBL).

1.2.1 Cacher son adresse courriel

Avant tout, la technique la plus simple pour éviter le spam consiste à protéger son adresse courriel des spammeurs. Ainsi, il ne faudra transmettre son adresse courriel qu'à des contacts de confiance. Pour ce qui est des contacts dont la confiance est moindre, il faudrait utiliser des adresses de courriel temporaire. Quand on inscrit son adresse sur un site web par exemple, il est préférable de rajouter un caractère ou

2. Simple Mail Transfer Protocol, abrégé SMTP, est le protocole généralement utilisé pour transmettre les courriels de l'émetteur vers les serveurs de messagerie. Les ports utilisés par ce protocole sont le 25 (sans authentification), 587 (avec authentification) et 465 (pour une connexion sécurisée SSL).

3. RFC ou Requests For Comments sont une série de documents officiels décrivant les aspects techniques d'Internet.

une balise à supprimer avant de vous écrire sur votre adresse courriel. Cette précaution empêche votre adresse courriel d'être récupérée par des robots qui collectent les adresses courriel sur le web.

Ces précautions sont souvent insuffisantes puisque l'adresse courriel peut être protégée par l'utilisateur mais ses contacts peuvent par inadvertance divulguer votre courriel à travers des fils de conversations par exemple. De plus, les robots qui récupèrent les adresses courriel sur le web sont de plus en plus performants et détectent souvent les balises que les utilisateurs rajoutent afin de protéger leur courriel.

1.2.2 Identification de l'émetteur par sa signature S/MIME

S/MIME (Secure / Multipurpose Internet Mail Extensions)[36] est une norme de cryptographie et de signature numérique de courriel encapsulée au format MIME. Elle permet d'assurer l'intégrité, l'authentification, la non-répudiation et la confidentialité des données. Le standard MIME permet d'inclure dans les messages électroniques des fichiers attachés autres que des fichiers textes (ASCII). C'est donc grâce à ce standard que l'on peut transmettre des pièces jointes de différents types.

S/MIME repose sur le principe de chiffrement à clé publique, il permet de chiffrer le contenu du message mais ne chiffrera pas la communication. La figure 1.1 illustre le processus de chiffrement et déchiffrement du message. Le message est tout d'abord crypté à l'aide de la clé publique du récepteur que l'émetteur possède déjà. Ensuite, il est transmis sous sa forme cryptée. Enfin à la réception du message, le récepteur utilise sa clé privée correspondante pour décrypter le message et pouvoir le lire.

S/MIME est difficile à mettre en œuvre puisque tous les clients de messagerie n'intègrent pas les signatures S/MIME. De même, son intégration aux webmails (solutions très utilisées par les utilisateurs) n'est pas réalisée dans la grande majorité des cas. De plus, la fonctionnalité va dépendre du navigateur utilisé. D'autre part, comme le message est crypté, s'il contient un programme malveillant celui-ci ne sera pas analysé par les serveurs puisque le message ne pourra être décrypté. Enfin, un

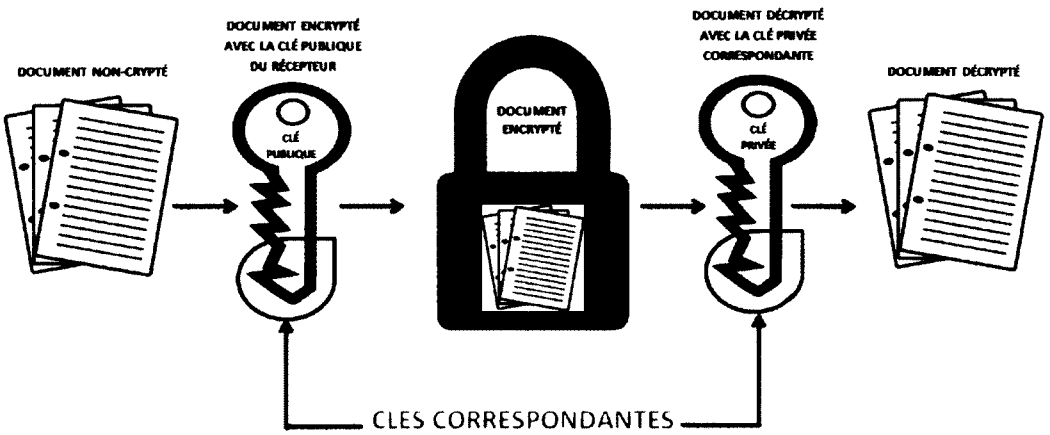

figure 1.1 – Exemple de chiffrement/déchiffrement avec clé publique

message encrypté par S/MIME ne pourra être décrypté si le récepteur a supprimé la clé de décryptage ou s'il possède une clé erronée.

1.2.3 L'authentification sur SMTP (SMTP\AUTH)

La version originale de SMTP (Simple Mail Transfer Protocol) spécifiée par Jon Postel dans les années 70 ne permettait pas d'utiliser un mot de passe pour transmettre un courriel. Depuis la fin des années 90, il est commun d'avoir des relais de courriels ouverts, ils sont considérés comme étant une faille. L'authentification SMTP [41], souvent appelée SMTP AUTH, est une extension du SMTP au moyen duquel le client SMTP peut s'identifier.

L'authentification SMTP est possible grâce aux serveurs le permettant, ils nécessitent que le client s'authentifie. Il faut en effet que le client et le serveur s'acceptent mutuellement en utilisant une méthode commune d'authentification. Au départ, ce protocole a été inventé comme un protocole serveur à serveur, avec l'authentification SMTP, un client doit s'identifier et seulement après la réussite de cette étape, la réception et la transmission de ses courriels sont effectuées.

1.2. Les différentes techniques basées sur le bloquage d'adresse IP utilisées dans la lutte contre le spam

L'authentification SMTP a été rajoutée aujourd'hui à la grande partie des serveurs SMTP, ce processus est défini dans une suite de RFC. La séquence suivante présente un échange type entre le serveur SMTP "S" et le client SMTP "C" :

```
S: 220 esmtp.example.com ESMTP
C: ehlo client.example.com
S: 250-esmtp.example.com
S: 250-PIPELINING
S: 250-8BITMIME
S: 250-SIZE 255555555
S: 250 AUTH LOGIN PLAIN CRAM-MD5
C: auth login
S: 334 VXNlcm5hbWU6
C: avlsdkfj
S: 334 UGFzc3dvcmQ6
C: lkajsdfvlj
S: 235 Authentication successful.
```

Nous pouvons voir dans l'échange ci-dessus les différentes étapes d'une authentification SMTP classique. Le client C lance une requête EHLO[4] au serveur. Le serveur va accepter la demande et attendre l'authentification du client ***250 AUTH LOGIN PLAIN CRAM-MD5***. Le client tente une connexion de type ***LOGIN***. Le serveur attend un nom d'utilisateur encodé en BASE64[5], le client va donner son nom d'utilisateur encodé ***avlsdkfj***. Ensuite, il s'agit de demander et transmettre le mot de passe dans le même encodage. Enfin, si les informations transmises par le client sont bonnes, alors la connexion se réalise ***235 Authentication successful*** et le message pourra ensuite être transmis au serveur SMTP.

Il existe trois mécanismes principaux d'authentification pour le SMTP\AUTH :

- AUTH LOGIN : les échanges d'informations sont encodés en BASE64.

4. La clause SMTP HELO est l'étape du protocole SMTP où les serveurs SMTP se présentent les uns aux autres. Le serveur d'envoi va s'identifier et le serveur de réception va accepter n'importe quel nom. EHLO est une version améliorée de HELO.

5. BASE64 est un codage d'information utilisant 64 caractères.

- AUTH PLAIN : le client transmet un message unique au serveur. Ce message n'est pas encodé et contient le nom d'utilisateur et le mot de passe d'authentification.
- AUTH CRAM MD5 : cette fois, l'échange des informations d'authentification sera encodé en CRAM-MD5, c'est-à-dire une combinaison d'un mécanisme de question/réponse et d'algorithme de hachage MD5[6].

Cependant SMTP/AUTH est une extension qui n'est pas présente sur tous les webmails et elle n'est pas non plus obligatoire, ce qui réduit son utilisation. De plus, SMTP/AUTH ne protège pas contre l'usurpation d'identité qui est une technique très utilisée par les spammeurs et qui consiste à utiliser une autre adresse IP que celle attribuée par défaut par les serveurs au terminal client. Cette technique permet au spammeurs d'éviter de se faire retracer par leur adresse IP et aussi d'utiliser cette adresse IP qui peut être une adresse IP dont les courriers vont être automatiquement acepter par les serveurs de transfert de courrier[7]

1.2.4 Authentification SMTPS

SMTPS [13] est une méthode qui sert à sécuriser une connexion SMTP de manière à offrir une authentification de l'émetteur, une intégrité et une confidentialité des données. SMTPS n'est pas un protocole propriétaire, ni une extension de SMTP, c'est juste une manière de sécuriser SMTP sur la couche transport du modèle OSI (voir annexe A). Ce qui signifie que le client et le serveur communiquent avec un SMTP standard sur la couche application, mais la connexion va être sécurisée par SSL ou TLS[8]. Cette authentification se fait avant l'envoi de toute donnée. Pour pouvoir bénéficier du protocole SMTPS, il faudra des serveurs qui acceptent des connexions SMTPS.

6. MD5 ou Message Digest 5 est une fonction de hachage cryptographique.

7. Les serveurs de transfert de courriel n'ont pas de boite réception, ils vont seulement transférer le courriel vers un autre serveur ou vers le récepteur.

8. SSL ou "Secure Sockets Layer" et TLS ou "Transoprt Layer Security" sont des protocoles cryptographiques qui permettent de sécuriser une communication sur Internet. Ils encryptent le segment de connexion sur la couche transport en utilisant une cryptographie asymétrique pour l'échange des clés, une encryption symétrique pour la confidentialité et les "messages authentification codes" (MAC) pour assurer l'intégrité du message

SMTPS ne protègera pas du usurpation du champ "De :". Effectivement, l'authentification au serveur SMTP à l'aide du protocole SMTPS et des identifiants n'empêchera pas l'émetteur de transmettre un courriel ayant un champ "De :" avec une fausse adresse courriel. De plus, les connexions SMTPS ne permettent pas l'utilisation des serveurs de transfert de courriel donc il y a aura des possibilités de retour de courriel ou des messages d'avertissement de non livraison du message. Enfin, les serveurs et webmails acceptant une connexion SMTPS sont peu nombreux.

1.2.5 Sender Policy Framework

Aujourd'hui, tous les courriers indésirables sont issus d'une fausse adresse. Les victimes dont les adresses ont été usurpées sont souvent placées par les destinataires sur liste noire. En effet, les protocoles Simple Mail Transfer Protocol (SMTP) utilisés pour transmettre les courriels ne prévoient pas de mécanisme de vérification de l'expéditeur, c'est-à-dire qu'il est facile de transmettre un courriel avec une adresse fausse ou usurpée. L'usurpation d'adresse est une menace pour les individus comme pour les entreprises ; elle altère également la confiance des utilisateurs pour l'utilisation de la communication par courriel. C'est pour cela, par exemple, que votre banque ne vous transmettra jamais de l'information concernant votre compte par courriel. D'où la mise en place du SPF (Sender Policy Framework) [21] qui est un système de validation du courriel créé pour prévenir le clonage d'adresse.

La vérification de la conformité avec SPF se réalise en trois étapes :

- Publier une politique : Les domaines et les hôtes doivent identifier les machines qui sont autorisées à transmettre des courriels en leur nom. Cela se fait en rajoutant des enregistrements aux informations DNS existantes. Par exemple, chaque domaine ou hôte qui a un "A record" ou "MX record" devrait avoir un SPF qui définit une politique si il est utilisé dans une adresse courriel ou un argument HELO/EHLO. Les hôtes qui ne transmettent pas de messages devraient publier un SPF qui indique "v=spf1 -all".
- Vérifier et utiliser les informations SPF : Les récepteurs utilisent des requêtes DNS classiques qui sont généralement mises en cache pour améliorer la perfor-

mance. Cette étape est cruciale puisqu'elle va permettre aux récepteurs d'interpréter l'information du SPF et d'agir en fonction du résultat.

- Réviser le transfert du courriel : Le transfert des courriels ordinaires n'est pas autorisé par SPF. Les alternatives sont : le "remailling" qui consiste à remplacer l'émetteur original par un autre qui appartient au domaine local, le "refusing" c'est-à-dire en répondant *551 Utilisateur non local; essayer avec <utilisateur@example.com>*, le "whitelisting" c'est-à-dire une liste blanche comme cela le serveur récepteur ne refusera pas un message transféré et le "Sender Rewriting Scheme" qui est un mécanisme plus compliqué qui s'occupe de faire parvenir les notifications de non-acheminement à l'émetteur original. Ainsi, le principal inconvénient de SPF réside dans la spécification des informations de DNS que le domaine définit et que les récepteurs utilisent.

Plus précisément, la version actuelle de SPF protège l'adresse enveloppe de l'expéditeur qui est utilisée lors de la transmission. L'adresse enveloppe est une adresse utilisée pour la communication entre deux serveurs de courriels. Elle est utilisée entre autres pour prévenir l'émetteur de l'échec d'un envoi et n'est pas visible par l'utilisateur. Elle est donc différente de l'adresse qui se trouve dans le champ destinateur qui est une adresse en-tête. SPF permet au propriétaire de l'adresse de spécifier une politique d'envoi du courriel, c'est-à-dire quel serveur d'envoi sera utilisé pour transmettre leur communication.

Cette technologie nécessite deux volets qui travaillent en même temps : d'une part, le propriétaire du domaine publie cette information dans un enregistrement SPF dans le domaine DNS ; d'autre part, quand le serveur d'un autre utilisateur reçoit un mail provenant possiblement du premier domaine alors il vérifie son authenticité en vérifiant si le courriel respecte la politique d'envoi du dit domaine. Une fois l'authenticité du destinateur confirmée, le courriel pourra être considéré comme légitime.

Cependant, les utilisateurs n'adoptent pas encore cette habitude. Bien au contraire, les spammeurs se crée des SPF plus rapidement. De plus, cette technique empêche l'utilisation des serveurs de transfert de courriel, ce qui a pour conséquence la non

délivrance de certain courriel.

Exemple :
Voici un exemple simple pour illustrer rapidement le fonctionnement des SPF. Bob est propriétaire du domaine example.net. Il utilise parfois son compte Gmail pour envoyer ses courriels et il a contacté le support technique de Gmail pour identifier l'enregistrement SPF que Gmail utilise. Puisqu'il reçoit souvent des retours de message qu'il n'a pas transmis, il décide de se créer un enregistrement SPF dans le but de réduire l'utilisation falsifiée de son domaine. Le tableau 1.1 présente le SPF qu'il va créer.

example.net. TXT "v=spf1 mx a :pluto.example.net
include :aspmx.googlemail.com -all"

v=spf1	SPF version 1
mx	les serveurs de réception (MXes) du domaine sont autorisés à transmettre un courriel pour example.net
a :pluto.example.net	la machine pluto.example.net est également autorisée
include :aspmx.googlemail.com	tout ce qui sera considéré légitime par gmail.com sera légitime pour example.net aussi
-all	toutes les autres machines ne sont pas autorisées à utiliser le domaine

tableau 1.1 – Exemple de SPF

1.2.6 DomainKeys Identified Mail (DKIM)

DomainKeys Identified Mail (DKIM) [1] qui est une fusion de "DomainKeys Authentification by Yahoo" et "Internet Identified Mail by Cisco" est une méthode d'authentification qui utilise le système de clé publique. DKIM est une authentification au niveau du domaine. Il permet donc d'associer un nom de domaine à un courriel

et donc de permettre à un domaine ou une organisation de revendiquer la responsabilité d'émission du message. L'association est mise en place grâce à une signature numérique qui peut être validée par le récepteur. La responsabilité de l'émission du message est revendiquée par le signataire en ajoutant une signature DKIM à l'en-tête du message. Le vérificateur va récupérer la clé publique du signataire en utilisant un DNS, et ensuite il va vérifier si la signature correspond avec le contenu du message comme présenté dans la figure 1.2. Une signature DKIM peut aussi contenir d'autres champs de l'en-tête du message comme les champs 'De :', 'Sujet :' et une partie du message lui-même. DKIM ne garantit pas l'intégrité du message.

Les avantages offerts par DKIM sont nombreux, il permet avant tout d'obtenir des 'blacklists' ou 'whitelists' de domaine plus efficaces puisqu'il permet au récepteur d'identifier de manière fiable un flux de courrier légitime. DKIM est une méthode qui permet donc d'attribuer un label à un message. Il ne permet pas directement de filtrer ou d'identifier un spam. Cependant, son utilisation généralisée peut empêcher les spammeurs d'usurper l'adresse source du message, une technique fréquemment utilisée par les spammeurs. Si les spammeurs sont obligés d'indiquer un domaine source correcte, d'autres techniques de filtrage vont alors être plus efficaces. Aussi, DKIM aide à identifier le courrier légitime qui ne devra pas être filtré, ainsi si un système de réception possède une liste blanche des domaines. Il pourra, par exemple, ainsi laisser passer le courrier signé par ce domaine et filtrer le reste de manière plus ou moins agressive.

Cependant, DKIM possède des faiblesses comme le fait que sa mise en place n'empêche pas un spammeur de composer un message à travers un domaine qui est autorisé pour obtenir une signature valide. Ainsi ce message pourra être transmis à plusieurs adresses sans réel contrôle. Le fournisseur de messagerie pourra bloquer le compte qui a transmis le courrier initial mais ne pourra pas stopper la diffusion des messages déjà signés. On peut limiter la diffusion de tel message en rajoutant un champ correspondant à un temps d'expiration.

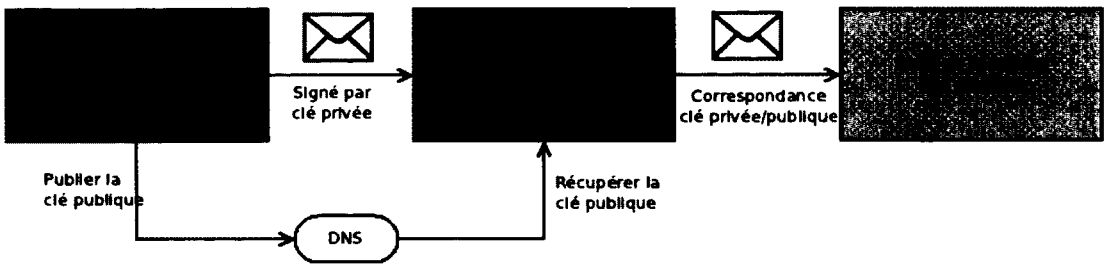

figure 1.2 – Fonctionnement de DKIM

1.2.7 Realtime Blackhole List (RBL)

Les RBL (Realtime Blackhole List) [26] sont une amélioration du système de liste noire classique d'adresses. Les listes noires classiques dépendent des administrateurs qui collectent et assemblent les adresses d'émetteurs de spams pour créer une "liste noire". Si l'émetteur du courriel est sur cette liste noire alors le message sera considéré comme du SPAM. Cependant, avec le développement de l'Internet, il est devenu impossible pour les administrateurs de créer une liste noire efficiente "manuellement". De ce fait, des listes noires en temps réel ont été créées et sont maintenues à jour grâce à un travail collaboratif à travers le réseau Internet.

Les RBL (Realtime Blackhole List, listes noires en temps réel) ont été créées par *"Mail Abuse Prevention System (MAPS) LLC."* mais il existe aujourd'hui d'autres fournisseurs de RBL. Ce processus consiste à lister les adresses IP des émetteurs de spams ou qui sont compromises en étant des relais de spam. Les FAI et les compagnies abonnées aux RBL vont pouvoir bloquer l'émission de courriels à partir de certaines adresses IP. Les RBL ou DNSBL (DNS-based Blackhole List) sont des listes d'adresses IP publiées via le Service de Nom de Domaine (DNS), les serveurs de courriel peuvent être configurés pour rejeter ou marquer les messages provenant d'une adresse présente dans une ou plusieurs de ces listes. La figure 1.3 présente le fonctionnement d'une RBL. On constate sur cette figure que le blocage se fait pendant la connexion SMTP ; l'instance d'authentification (le FAI) vérifie auprès des RBL la présence de l'adresse IP de l'émetteur, si une correspondance est trouvée alors la connexion sera refusée. Les 6 étapes principales présentées dans la figure sont les suivantes :

1. L'émetteur fait une requête d'authentification auprès du Fournisseur d'Accès Internet (FAI).
2. Le FAI vérifie auprès des RBL la présence de l'adresse IP de l'émetteur.
3. Si l'adresse n'est pas répertoriée dans les RBL,
4. l'émetteur du courriel est authentifié.
5. L'émetteur transmet son courriel au FAI.
6. Le FAI délivre le courriel au récepteur.

Les listes sont constituées effectivement des adresses IP à partir desquelles proviennent les spams. Dans la plupart de ces cas, cela n'affecte pas seulement le spammeur mais aussi tous les clients du FAI puisqu'ils ont le même fournisseur d'accès internet. Il est alors demandé au FAI de débrancher le spammeur pour ne plus apparaitre sur la RBL. Mais la raison la plus fréquente de l'apparition d'une adresse IP sur la RBL est due au fait que l'adresse IP soit utilisée comme un relais d'envoi de spam. Les spammeurs utilisent les serveurs de courriels qui sont configurés pour transférer tous les courriels qu'ils reçoivent même ceux qui ne leur sont pas destinés. De ce fait, bloquer les courriels pour ce type de serveurs peut être une mesure très complexe à mettre en place. En effet, si le relais lui-même est corrompu, il va se bloquer lui-même et plusieurs courriels légitimes ne seront pas transmis.

Nous pouvons constater que la probabilité d'être injustement mis sur liste noire et de voir donc ses correspondances interrompues pour aucune raison précise est trop importante. De ce fait, cette technique basée sur les listes noires d'adresses IP donne de très mauvais résultats au niveau des faux positifs. De plus, vu que la plupart des spams sont aujourd'hui transmis à travers de fausses adresses IP d'émetteur, les chances que la RBL ne détecte pas le spam sont considérables. De ce fait, cette technique est souvent jumelée à d'autres.

1.2.8 Le greylisting

Le greylisting [27] est une technique antipourriel (antispam) particulièrement efficace, qui fonctionne selon le principe que lorsqu'un serveur de réception de courriel

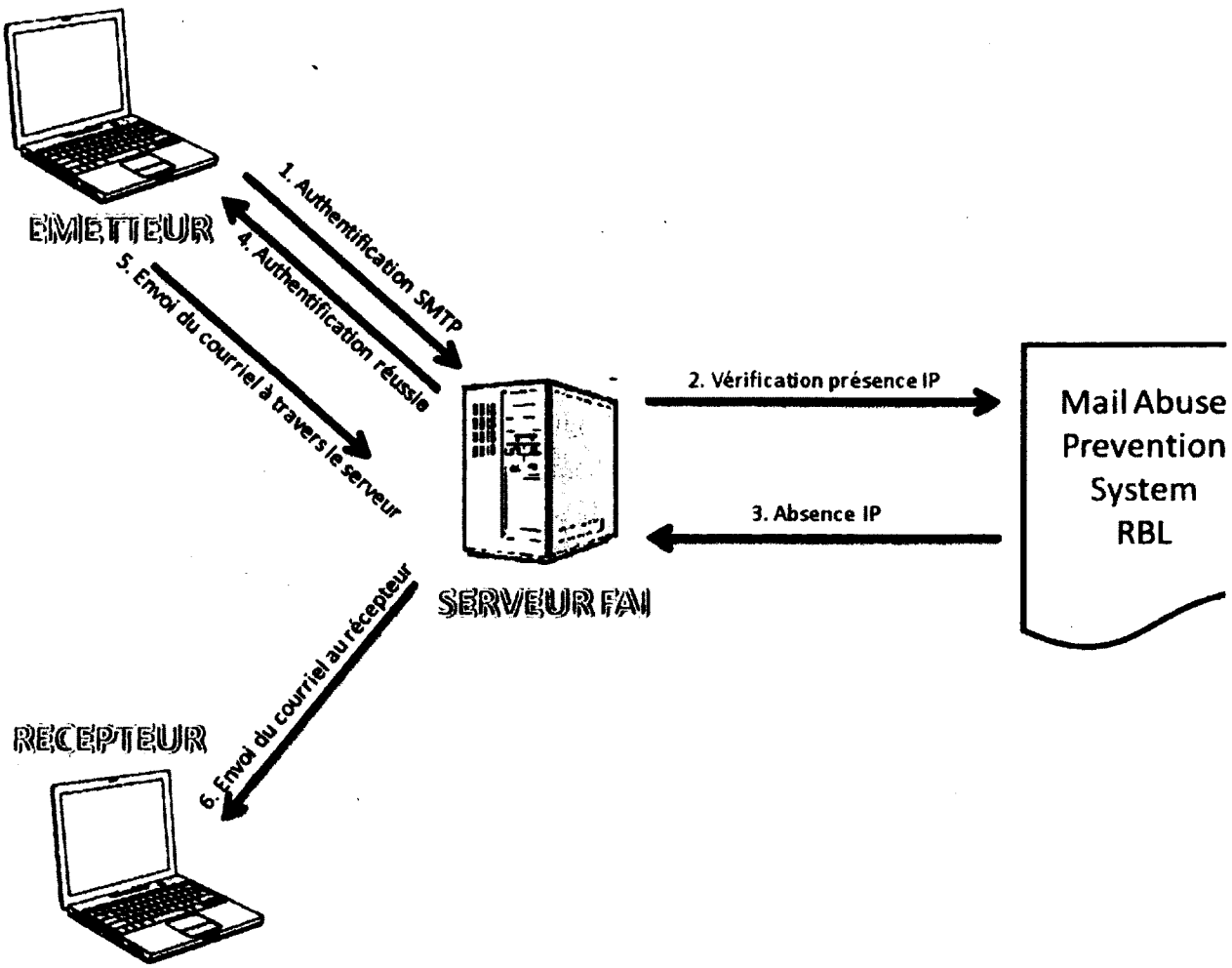

figure 1.3 – Fonctionnement d'une vérification par RBL

(dans ce cas-ci, le serveur qui reçoit le courriel, sur lequel le filtre de courriel est activé) reçoit un courriel et qu'il ne peut traiter la réception d'un message (par exemple, s'il est indisponible), il doit retourner un code d'erreur 421. Ce code d'erreur indique au serveur qui envoie le message d'attendre et de réessayer l'envoi un peu plus tard. Ce délai est défini dans la configuration du serveur expéditeur du message (ou Mail

Transfert Agent : Mail Transfert Agent). Les MTA légitimes respectent cette règle. Les MTA non légitimes (utilisés par les polluposteurs) ne le font pas, car cela leur fait perdre de l'efficacité : le MTA continue donc son envoi de courriels (il passe au prochain destinataire) sans attendre pour réenvoyer le courriel actuel. On utilise donc cette particularité pour créer une "liste grise". Celle-ci fonctionne avec une base de données. Chaque enregistrement de la base de données constitue un triplet composé de l'adresse IP du serveur qui envoie le courriel, de l'adresse courriel de l'expéditeur, et de l'adresse courriel du destinataire, formant ainsi une clé unique. Lorsqu'un message est reçu par le serveur de courriel du destinataire, ce dernier vérifie dans sa base l'existence du triplet. À partir de là :

- Si le triplet n'est pas dans sa base de données, il l'ajoute avec la date actuelle. Il renvoie ensuite le code d'erreur 421, indiquant au serveur qu'il devra réenvoyer le message.
- Si le triplet est déjà dans la base de données, le serveur vérifie le délai entre la date courante et celle stockée dans la base (la date de la première connexion).

Si le délai est supérieur ou égal au délai prédéfini, le message est accepté. Sinon, le serveur retourne un numéro d'erreur 421. Après un certain temps (défini également dans l'enregistrement), l'enregistrement devient inactif et le serveur doit réenvoyer un 421. Ainsi, lorsque le MTA envoyeur reçoit le 421, s'il est légitime, il attendra avant de ré-envoyer le message. Sinon, il n'attendra pas et ne le réenverra pas. Cette technique permettait d'atteindre des taux d'efficacité très élevés, de l'ordre de 99 % quand elle a été proposée en 2003, puisque la très grande majorité des polluposteurs préfère sacrifier un courriel plutôt que d'attendre et ainsi, diminuer leur performance. Actuellement, l'efficacité est moins importante (80-90 %) à cause de l'augmentation de l'utilisation des webmails (de vrais serveurs de messagerie), par les polluposteurs, pour distribuer les pourriels.

Cette technique crée également des retards dans la communication et augmente le nombre d'échanges sur le réseau puisque pour l'envoi d'un courriel par une IP qui n'est pas encore enregistré auprès du destinataire il y aura dans le meilleur des cas 3 communications plutôt qu'une ; l'envoi initial, la réponse 421 et enfin le renvoi dans les délais.

1.2.9 Filtrage heuristique

Le filtrage heuristique [12] fonctionne en soumettant les courriels à plusieurs tests prédéfinis pour vérifier l'enveloppe, l'en-tête et la structure du contenu du message. Chaque règle attribuera un score numérique quand à la probabilité que le message soit un spam. Le résultat de la somme de ces scores est appelé "Spam Score".

Le "Spam Score" est alors comparé à la sensibilité configurée par l'utilisateur. Selon cette sensibilité plus ou moins de spams vont être détéctés, mais à contre-mesure, il y aura plus de risque qu'un courrier légitime soit considéré comme du spam.
Selon le "Spam Score" et la sensibilité, les messages seront donc classifiés en spam, non-spam, et indécis. Il y a des exceptions à cette règle :

- les messages en provenance d'émetteurs sur la liste des contacts approuvés ne seront jamais considérés comme du spam.
- à l'inverse, les messages en provenance d'émetteurs non approuvés seront toujours considérés comme du spam.
- enfin, les messages respectant des règles prédéfinies et personnalisées par le récepteur ne seront pas traités comme du spam.

Le problème majeur de ces machines heuristiques est que les polluposteurs apprennent et s'adaptent à ces règles heuristiques.

1.3 Tableau récapitulatif

Techniques	Avantages	Inconvénients

1.3. Tableau récapitulatif

Techniques	Avantages	Inconvénients
Cacher son adresse courriel	- Réduit la quantité de spams reçus	- Les contacts peuvent ne pas contribuer à cette attention en incluant par exemple votre adresse dans une liste de diffusion - Les robots apprennent à détecter et supprimer les tags rajoutés pour protéger l'adresse courriel
Signature S/MIME [36]	- Permet d'assurer l'intégrité, l'authentification, la non-répudiation et la confidentialité	- Tous les clients de messagerie et tous les webmails n'intègrent pas cette technique - Si le courriel contient un code malveillant, il ne sera pas détecté par les serveurs puisqu'ils ne pourront décrypter le courriel - S'il y a corruption d'une clé le message ne sera pas récupérable
SMTP/AUTH [41]	- Permet de réaliser une authentification sécurisée entre l'émetteur et le serveur d'envoi de courriel	- Extension non obligatoire et très peu utilisée par les webmails - Usurpation d'identité de l'adresse IP et du champ "De :" possible

1.3. Tableau récapitulatif

Techniques	Avantages	Inconvénients
STMPS [13]	- Permet de réaliser une authentification sécurisée entre l'émetteur et le serveur d'envoi de courriel - Permet de garantir l'intégrité et la confidentialité du message	- Les serveurs proposant ce protocole sont peu nombreux - Ne passe pas au travers des serveurs de transfert de courriels - Usurpation d'identité du champ "De :" possible
SPF [21]	- Permet de vérifier l'authenticité de l'adresse courriel d'envoi - Protège l'adresse de provenance	- Les spammeurs adoptent SPF plus rapidement que les courriers légitimes - Ne passe pas au travers des serveurs de transfert de courriels
DKIM [1]	- Permet indirectement de constituer des "blacklists" et "whitelists" de domaine	- Un courriel transmis et dont on réalise la corruption ne pourra être stoppé et sera automatiquement diffusé par les serveurs de transfert de courriel
RBL [26]	- Permet un blocage ou marquage des courriels provenant des adresses IP de listes noires	- Le blocage peut affecter tous les utilisateurs du serveur si le serveur est sur une "blacklist" puisqu'il a transmis du spam

1.3. TABLEAU RÉCAPITULATIF

Techniques	Avantages	Inconvénients
Greylisting [27]	- Méthode adaptative - Permet de bloquer selon des caractéristiques bien précises	- Les spammeurs s'adaptent aussi et changent les caractéristiques des spams - Il y a un temps d'adaptation et tous les spams ne seront pas bloqués

tableau 1.2: Tableau récapitulatif des avantages et inconvénients des techniques de lutte contre le spam

Le tableau 1.2 fait un récapitulatif des avantages et inconvénients des techniques et architectures utilisées pour lutter contre le spam. Celles-ci sont souvent combinées entre elles pour palier à leur manque. Malgré cela les spammeurs contournent ces restrictions et transmettent du spam. Ainsi, ces techniques vont être également combinées avec des méthodes qui se basent sur une classification du contenu du message. Nous explorons l'approche basée sur un classificateur bayésien dans le chapitre suivant puisque c'est le classificateur le plus répandu dans le domaine.

Chapitre 2

La lutte anti-spam basée sur les filtres bayésiens

Nous pouvons constater que les techniques basées sur le blocage d'adresse IP et le filtrage en utilisant les particularités des protocoles, tel que SMTP, impliquent un investissement de la part des FAI et peuvent entraîner très rapidement un manque d'efficacité. En effet, ces techniques sont universelles aux clients du FAI et ne sont donc pas personnalisées ni personnalisables.

Jusqu'à la fin des années 80, on utilisait des classificateurs [1] basés sur des règles définies par l'utilisateur. Définir des classificateurs performants et précis était difficile et prenait beaucoup de temps puisqu'il fallait définir des classificateurs pour chaque cas particulier. C'est avec les progrès technologiques et l'augmentation des performances des machines qu'on a commencé à utiliser des classificateurs statistiques. Ces derniers utilisent l'apprentissage à travers des données d'entraînement dont on connaît déjà la classification pour construire le classificateur.

Nous allons dans ce chapitre présenter des filtres basés sur des données statistiques. Ces filtres sont généralement évolutifs. Ils ont pour but de limiter la réception de spams (c'est-à-dire ne les arrêtent pas complètement) tout en évitant les faux positifs,

1. La tâche d'un classificateur consiste grossièrement à attribuer à un motif donné une classe

c'est-à-dire de détecter un courrier légitime comme étant un spam. Dans ce chapitre, nous étudions les techniques utilisées dans ce domaine. La genèse de ces techniques de filtrage a été marquée par un article central "A plan for spam" de Paul Graham en 2002 [11] dans lequel il présente l'utilisation d'un filtre bayésien naïf pour détecter les spams.

2.1 Apprentissage statistique

2.1.1 Définition

Avant de parler de notions en rapport avec la statistique bayésienne, il faut prendre conscience de la notion d'apprentissage statistique et de son poids dans l'efficacité des algorithmes que nous allons rencontrer durant notre étude. En effet, l'abondance des données rencontrées dans les analyses statistiques couplée à leur complexité rend l'exploitation de ces données par des moyens humains de plus en plus complexe. De ce fait, on met en place des méthodes automatiques d'analyse qui confèrent aux algorithmes des capacités de fonctions de perception, discrimination et reconnaissance basées sur un modèle humain malgré nos connaissances limitées dans le domaine. C'est ce qu'on appelle l'apprentissage et il est basé sur une base de connaissances que l'on continue à enrichir avec de nouvelles données.

Il existe deux grandes approches d'apprentissage [3] [44] : l'apprentissage non supervisé où l'algorithme se doit d'être totalement autonome et l'apprentissage supervisé qui va être corrigé et maintenu par un facteur extérieur. C'est ce dernier modèle qui nous intéresse puisqu'on peut corriger les erreurs de classification de nos algorithmes et de ce fait même améliorer sa base de connaissances dans le but d'obtenir de meilleurs résultats et s'adapter au changement. Ainsi, le but principal de la théorie d'apprentissage statistique est de fournir un cadre d'étude du problème d'inférence afin de pouvoir améliorer sa base de connaissances, faire des prédictions, prendre des décisions ou construire un modèle à partir d'un ensemble de données. On décrit le modèle général de l'apprentissage d'un exemple de trois composantes :

1. un générateur (G) de vecteurs aléatoires $x \in R^n$ construit de façon indépendante à partir d'une fonction de distribution de probabilité $F(x)$.
2. un superviseur (S) qui retourne une valeur de sortie y pour tous les vecteurs x entrés selon fonction de distribution conditionnelle $F(y|x)$.
3. une machine d'apprentissage (LM) capable d'implémenter un ensemble de fonctions $f(x, \alpha)$, $\alpha \in \Lambda$, où Λ est un ensemble de paramètres.

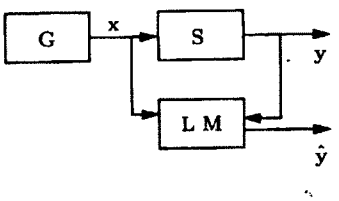

figure 2.1 – Un modèle d'apprentissage à partir d'exemples

L'exemple de la figure 2.1 présente un cas dans lequel durant le processus d'apprentissage, la machine d'apprentissage (LM) observe la paire (x, y) (l'ensemble d'apprentissage). Après l'apprentissage, la machine doit retourner quelque soit le x donné une valeur $\bar{y}$. Le but est de retourner une valeur de $\hat{y}$ qui sera proche de la réponse y du superviseur [43].

Le problème de l'apprentissage est celui de choisir à partir de l'ensemble des fonctions $f(x, \alpha)$, $\alpha \in \Lambda$, celui qui donne une meilleure approximation de la réponse du superviseur. La sélection de la fonction désirée est basée sur un ensemble d'apprentissage de l observations indépendantes et identiquement distribuées établies selon $F(x, y) = F(x)F(y|x)$.

2.1.2 Sur-apprentissage (Overfitting)

L'overfitting apparaît quand le modèle statistique décrit une erreur ou un bruit au lieu de le supprimer sans le considérer. Ce problème apparaît généralement quand le

problème contient, par exemple, un trop grand nombre de paramètres. Ce modèle aura des résultats très mauvais dans la prédiction. Il y a possibilité d'overfitting lorsque :

- si l'erreur de prédiction sur l'ensemble d'apprentissage diminue alors que l'erreur sur la validation augmente de manière significative. Cela signifie que le réseau continue à améliorer ses performances sur les échantillons d'apprentissage mais perd son pouvoir de prédiction sur ceux provenant de la validation.
- le modèle mémorise l'ensemble d'apprentissage au lieu d'apprendre à généraliser à partir de ces données.

Généralement, un algorithme d'apprentissage est entraîné en utilisant un ensemble de données de test, c'est-à-dire des exemples de situations dont on connaît la sortie. La machine d'apprentissage doit atteindre un état à partir duquel elle pourra prédire une sortie correcte pour des entrées non connues. Cependant comme sur la figure 2.2, lorsque l'apprentissage est trop long (c'est-à-dire que l'erreur de prédiction sur l'ensemble d'apprentissage diminue alors que l'erreur sur la validation augmente de manière significative) ou que les exemples d'apprentissage sont trop rares, l'algorithme va s'ajuster aux caractéristiques aléatoires très particulières de l'ensemble d'apprentissage et qui n'apporte pas de critère de prédiction fiable.

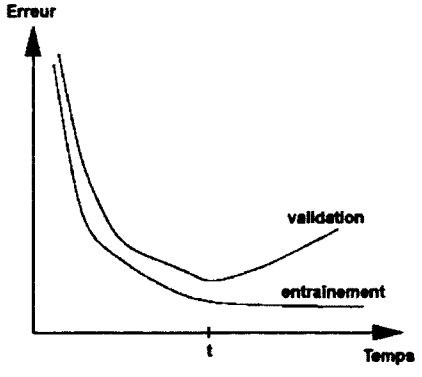

figure 2.2 – Erreur d'apprentissage en fonction du temps

2.2 Modèle général

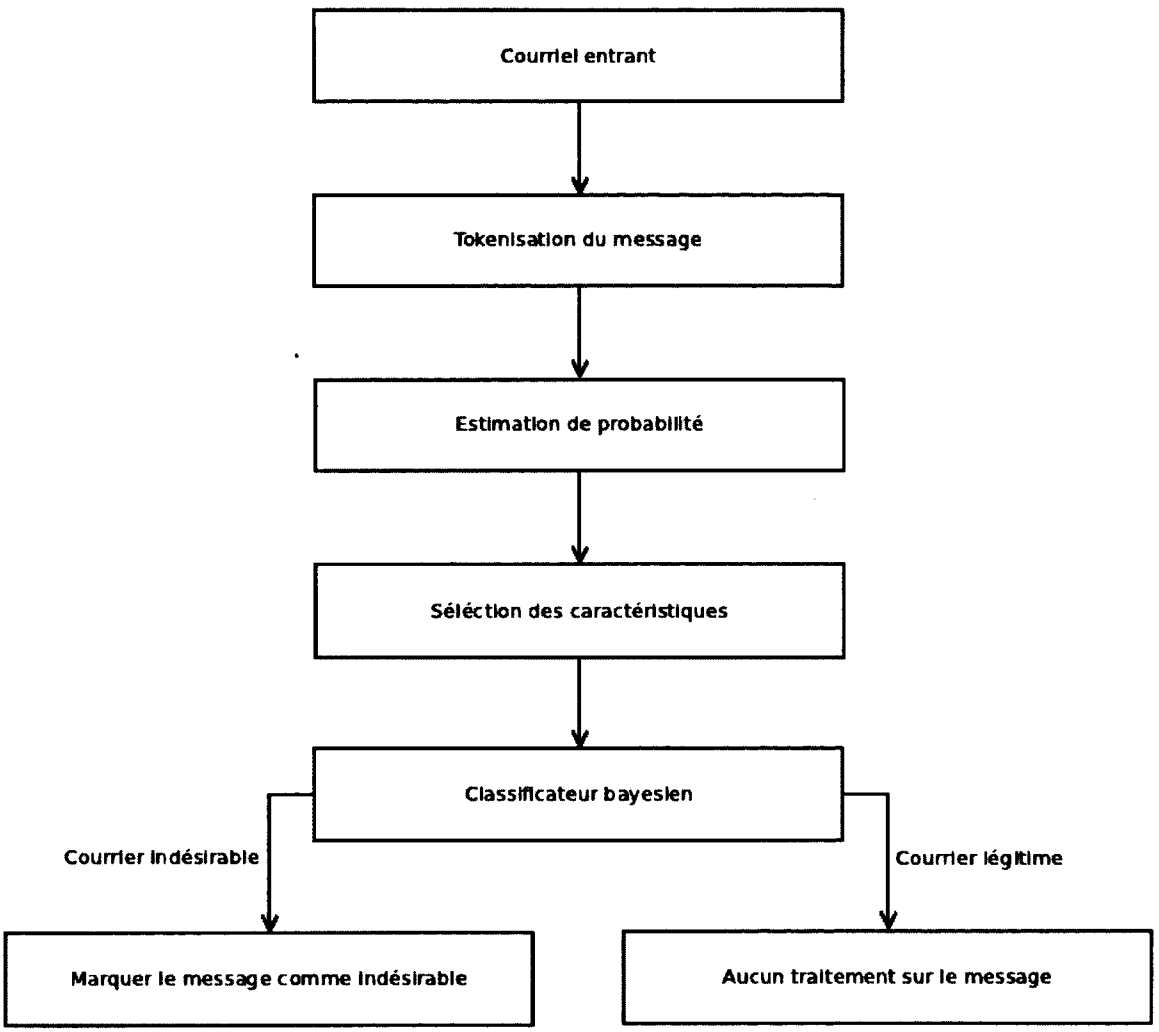

figure 2.3 – Modèle de filtre bayésien naïf

Le traitement d'un nouveau courriel est décrit par la figure 2.3. Quand un message arrive, il est d'abord découpé dans un ensemble de caractéristiques (en token). Ensuite, on attribue à chaque caractéristique une estimation de probabilité sur le fait qu'il représente un courrier indésirable ou non en considérant la base de connaissances. Par la suite, pour réduire la dimensionnalité du vecteur de caractéristiques, on procède à une sélection de caractéristiques. Enfin, le classificateur bayésien naïf

va combiner les probabilités de toutes ses caractéristiques retenues pour définir la probabilité que le message soit ou non un courrier indésirable. Si c'est un courrier indésirable alors il faudra le traiter en marquant le message comme étant indésirable. On utilise un classificateur bayésien naïf pour simplifier en faisant l'hypothèse bayésienne naïve que les caractéristiques sont indépendantes.

Nous allons voir les différentes étapes de ce modèle et constater l'importance de chacune des étapes sur la classification finale.

2.2.1 Quelques définitions

L'**alphabet** $A = \{a_1, a_2, ..., a_n\}$ est un ensemble fini de symboles. Chaque symbole est un caractère. Par exemple, l'alphabet français $F = \{a, b, ..., z\}$ ou l'alphabet binaire $B = \{0, 1\}$. Les courriels utilisent les caractères ASCII, $A_{ASCII} = \{char(0), char(1), ..., A, B, ..., char(255)\}$, avec $char(i)$ le caractère correspondant à l'entier i tel que $0 \leq i \leq 255$.

Une **chaîne de caractères** X_A est une séquence de caractères de l'alphabet A notée $X_A =< s_1, s_2, ..., s_n >: s_i \in A$. Chaque courriel est une chaîne de caractères de l'alphabet ASCII A_{ASCII}. Une caractéristique est représentée par une chaîne de caractères

La **cardinalité** d'une chaîne de caractères X est notée $|X| = n$. Par exemple, si X = <spam> alors $|X| = 4$. Ainsi la cardinalité de l'alphabet ASCII est $|A_{ASCII}| = 256$.

Un **ensemble de délimiteurs** D d'une chaîne de caractères X d'un alphabet A est un sous-ensemble de A. Par exemple, pour la chaîne de caractères X = $< Comment$ *ça va*? $>$, on pourra choisir l'ensemble de délimiteurs $D = \{espace,$ *point d'intérrogation*$\}$.

Un **"tokenizer"** T est une fonction d'une chaîne de caractères X et d'un ensemble de délimiteurs D et qui retourne un ensemble de chaines de caractères in-

clues dans X ainsi $Q = T(X, D)$. Par exemple, pour la chaîne de caractères $X =<$ *Comment ça va?* $>$ et l'ensemble de délimiteurs $D = \{espace\}$, on aura $Q = T(X, D) = \{Comment, ça, va?\}$.

Un **ensemble de libellés** $L = l_1, ..., l_m$ est un ensemble fini de libellés l_i avec $1 \leq i \leq m$. L'ensemble de libellés pour les courriels est $L = \{légitime, spam\}$.

Un **classificateur** C est une fonction qui associe un message X à un label l.

2.2.2 Tokenisation

Cette étape consiste à découper le message en choisissant un délimiteur. Ce choix de délimiteur influe considérablement sur la précision de la classification. Il existe deux méthodes de recherche de délimiteurs : "filter" et "wrapper".

Le choix des délimiteurs doit maximiser le rappel tout en conservant une précision élevée dans la classification. En effet, une des raisons de la baisse de la précision vient du fait que plus on a de délimiteurs, plus on perd de l'information. Ce phénomène contribue à la destruction des motifs discriminatifs qui permettent de classifier le message. L'interaction entre délimiteurs est complexe. Nous présentons dans ce qui suit la non-transitivité ainsi que la non-monotonie de cette interaction.

Soit un alphabet $A = \{a, b, c, d\}$ et les messages $X =< abcd >$ et $Y =< bcad >$. Soit X un message légitime et Y un polluriel. On définit D comme délimiteurs et la fonction $J(D) = \|Q_1 \cup Q_2\| - \|Q_1 \cap Q_2\|$ est définie en comptant le nombre d'éléments différents dans Q_1 et Q_2.

Un ensemble de délimiteurs qui produit des chaînes de caractères différents quand un message est un polluriel ou non est considéré comme bénéfique au contraire d'un tokeniseur qui ne le permet pas.

D	Q_1	Q_2	$J(D)$
$\{a\}$	$\{bcd\}$	$\{bc, d\}$	3
$\{a, b\}$	$\{cd\}$	$\{c, d\}$	3
$\{b, c\}$	$\{a, d\}$	$\{ad\}$	3
$\{a, c\}$	$\{b, d\}$	$\{b, d\}$	0

tableau 2.1 – Illustration de la relation de non-transitivité entre délimiteurs

Non-transitivité

On définit $\rightarrow$ une relation entre deux ensemble de délimiteurs. Une relation de transitivité peut s'écrire comment suit : $a \rightarrow b \wedge b \rightarrow c \Longrightarrow a \rightarrow c$. Grâce au tableau 2.1, on peut dire que les délimiteurs $\{a, b\}$ et $\{b, c\}$ sont bénéfiques puisqu'ils donnent des ensembles de tokens dissociées. S'il existait une relation de transitivité alors on aurait l'ensemble $\{a, c\}$ qui serait aussi bénéfique ; or, ce n'est pas le cas. Donc l'interaction entre délimiteurs n'est pas transitive.

Non-monotonie

La propriété de monotonie permet d'assurer qu'à l'ajout d'un nouveau délimiteur, on améliore la fonction $J(D)$. On note cette propriété comme suit : soient deux ensembles D_1 et D_2, $D_1 \subset D_2 \Rightarrow J(D_1) < J(D_2)$. Or dans l'exemple précédent, $\{a\} \subset \{a, c\} \not\Rightarrow J(\{a\}) < J(\{a, c\})$. Donc l'interaction entre délimiteurs est non monotone.

Filters et Wrappers

Les deux méthodes de sélection de tokens (qui sont les caractéristiques) les plus utilisées sont les méthodes d'induction "wrappers" et "filters".

L'approche "filters" [19] n'utilise pas l'induction pour estimer la pertinence des caractéristiques, elle utilise seulement la distribution des données. Donc, les filtres ne seront pas affectés par les erreurs de l'inducteur. Cette méthode est plus rapide que la méthode "wrappers". La figure 2.4 représente une méthode "filters".

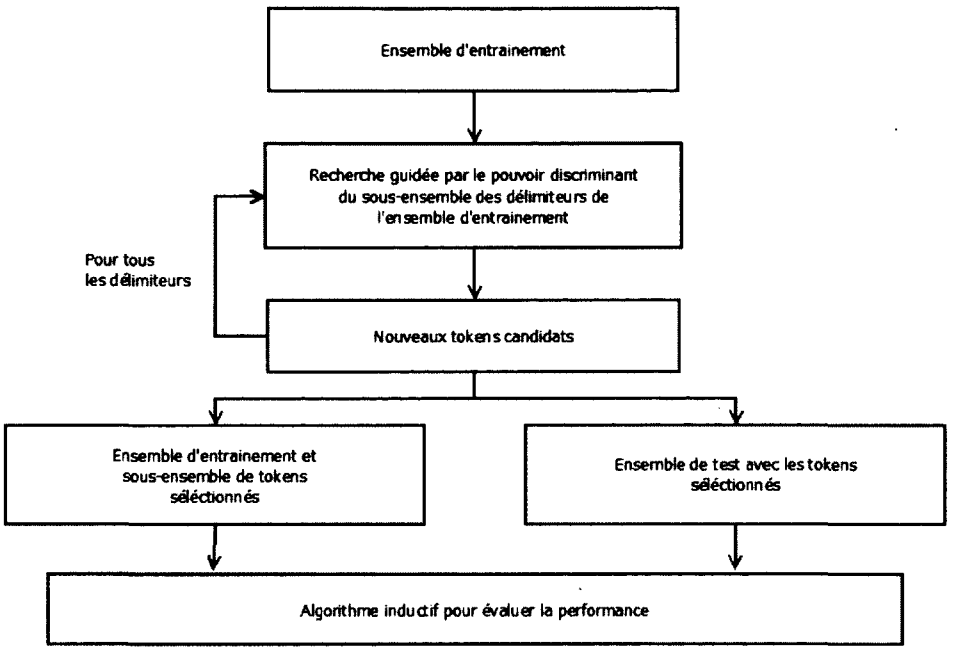

figure 2.4 – Illustration de la méthode "filters"

L'approche par induction "wrapper" [19] [22] [23] utilise l'inducteur dans la fonction de pertinence. L'induction "wrappers" est utilisée dans l'extraction de l'information sur Internet comme pour les marchés des actions ou les catalogues de produits. Elle est utilisée pour trouver par exemple des délimiteurs dans un code HTML pour extraire la meilleure information. La figure 2.5 représente la méthode "wrappers". Cette méthode donne généralement de meilleurs résultats que la méthode "filters".

2.2.3 Estimation de probabilité

L'approche la plus directe pour estimer la probabilité d'un token est de diviser son nombre d'occurrences par le nombre total de tous les tokens. Plus l'échantillon est grand et plus cette estimation approchera la probabilité théorique. Si toute la population était disponible, on aurait obtenu l'exacte probabilité du token. Mais généralement quand l'échantillon est de taille finie, l'estimation du maximum de vraisemblance *(MLE, Maximum Likelihood Estimate)* sera dans le voisinage de la probabilité.

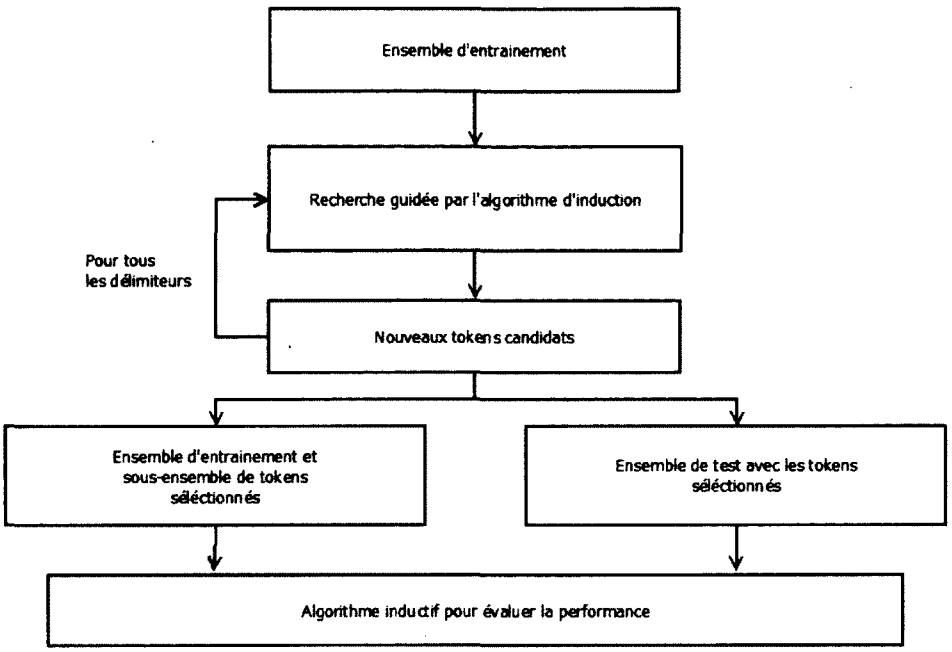

figure 2.5 – Illustration de la méthode "wrappers"

$$P_{MLE}(x_i) = \frac{|x_i|}{N}$$

Avec x_i le token et N le nombre total d'instances d'entraînement. Le problème de cette estimation est qu'elle attribue une probabilité nulle aux tokens qui n'ont pas encore été découverts. On appelle ce phénomène le problème de zéro-fréquence. Dans le cas d'un classificateur bayésien naïf (voir plus loin), où les estimations sont combinées à partir de deux corpus, une estimation d'un token qui serait égale à zéro pourrait entraîner le fait que la probabilité générale que cela soit un spam soit égale à 0. Il existe aussi le problème pour les tokens rares, ceux avec une fréquence basse. Par exemple, un mot qui est présent une fois dans le corpus des courriers légitimes et 10 fois dans l'autre est moins fiable qu'un mot qui va être présent 10 fois dans le corpus des courriers légitimes et 100 fois dans le corpus des spams. Même si les probabilités sont égales, le dernier token sera plus fiable.

Il existe des solutions concernant le problème de MLE pour des cas avec des données clairsemées. On applique la méthode de lissage qui consiste à déplacer le poids de la probabilité des éléments déjà rencontrés vers les éléments qui n'ont encore jamais été rencontrés.

Estimation absolue

L'estimation absolue [34] soustrait une constante aux fréquences

$$P_{Abs}(x_i) = \frac{|x_i| - c}{N}$$

c est une constante généralement obtenue à partir des fréquences dde tokens rencontrés juste une fois pour N_1 et deux fois pour N_2. Ainsi :

$$c = \frac{N_1}{N_1 + 2N_2}$$

Estimation de Laplace

L'estimation de Laplace est une technique de lissage qui part du principe que tous les événements ont été rencontrés au moins une fois avant.

$$P_{Lap}(x_i) = \frac{|x_i| + 1}{N + B}$$

Avec B le nombre de tokens différents. Le problème de cette estimation est qu'elle attribue trop de poids aux tokens qui n'ont encore jamais été rencontrés (x_i) [9].

Loi de Jeffreys-Perks

Cette méthode attribue moins de poids aux tokens non encore rencontrés. Ainsi quand l'estimation de Laplace attribue 1, cette méthode attribue un poids de 0.5.

$$P_{JP}(x_i) = \frac{|x_i| + 0.5}{N + 0.5B}$$

Estimation de Lidstone

Laplace et Jeffreys-Perks sont deux cas particuliers de l'estimation de Lidstone qui ajoute δ à chaque occurrence.

$$P_{Lid}(x_i) = \frac{|x_i| + \delta}{N + \delta B}$$

Avec $0 \leq \delta \leq 1$. Cette estimation est obtenue de la même manière que les deux méthodes précédentes à la différence que cette méthode n'attribue pas une constante mais elle utilise une variable ajustable δ. Le problème va être de trouver une bonne valeur de δ.

Lissage de Witten Bell

À l'origine, cette méthode [45] a été développée pour des besoins dans la compression de texte. Elle utilise les items rencontrés une fois pour estimer les items non encore rencontrés. La probabilité attribué à ces derniers est :

$$P(x_i) = \frac{N_1}{N_1 + N} \text{ pour } |x_i| = 0$$

N_1 est le nombre d'items rencontrés une seule fois. Pour les autres items, on aura :

$$P_{wb}(x_i) = \frac{x_i}{N + N_1} \text{ pour } |x_i| > 0$$

Estimation de Good Turing

L'estimation de Good Turing [10] utilise l'équation suivante pour calculer la probabilité des événements qui ont déjà été observés :

$$P_{GT}(x_i) = \frac{r^*}{N}$$

$$\text{avec } r^* = (r+1).\frac{E(N_{r+1})}{E(N_r)}$$

E(n) estime combien de mots différents ont été rencontrés n fois, r est la fréquence de l'item, N_r est la fréquence de cette fréquence et r^* est la nouvelle fréquence estimée. Il existe plusieurs estimations de Good Turing elles vont dépendre du calcul de E.

2.2.4 Sélection de caractéristiques

L'intérêt de réaliser une sélection de caractéristiques (c'est-à-dire une sélection de tokens qu'on a créé à l'étape de tokenisation et dont on a calculé ensuite l'estimation de probabilité) est de réduire la dimensionnalité ayant pour but un gain de temps d'exécution. C'est un aspect très important dans le domaine de la classification de textes où la grande dimensionnalité des caractéristiques représentent un réel problème. Dans la littérature plusieurs méthodes ont été proposé pour réaliser cette tâche : χ^2, Information Gain, Mutual Information, Term Strength, Document Frenquency, Odds-Ratio, etc. Une étude comparative de ces méthodes a été réalisé par Yang et Pedersen [46] qui montre que χ^2, Information Gain et Document Frequency donnaient les meilleurs résultats dans un corpus comme celui de Reuters en utilisant le classificateur KNN (k-Nearest Neighbor). Une autre étude de Mladenic et Grobelnik [32] montre que Odds-Ratio obtient les meilleurs résultats avec un classificateur bayésien. Les méthodes χ^2, Information Gain et Probability Ratio sont les trois méthodes communément utilisées dans la littérature dans la classication du spam.

2.3 Filtre bayésien

2.3.1 Introduction à la statistique bayésienne

La probabilité a priori $P(A)$ d'un événement A est la probabilité que A survienne sans prendre en considération aucun autre événement. La probabilité conditionnelle d'un événement est la probabilité que l'événement survienne en sachant qu'un autre événement est survenu.

La théorie de la décision bayésienne est basée sur l'hypothèse que le problème de décision est posé dans un contexte probabiliste et que toutes les probabilités pertinentes sont connues [8]. Pour mieux comprendre le processus de décision, nous allons prendre l'exemple d'un ensemble de planche de pins et sapins découpés dans une forêt. Soit ω *l'état de la nature, c'est-à-dire la proportion du type d'arbre donc de planche dans cette forêt*, avec $\omega = \omega_1$ pour une planche de pin et $\omega = \omega_2$ pour une planche de sapin. Comme on ne connait pas la proportion de chacun des types de planche dans

la nature, nous considérons ω comme une variable aléatoire.

Si la forêt produisait autant de pins que de sapins, nous aurions conclu que les chances que la prochaine planche soit du pin ou du sapin auraient été égales. Plus généralement, nous considérons qu'il existe une certaine probabilité a priori que la prochaine planche soit du pin $P(\omega_1)$ ou qu'il soit du sapin $P(\omega_2)$. Ces probabilités a priori qui sont non nulles avec $P(\omega_1) + P(\omega_2) = 1$, représentent nos connaissances a priori de rencontrer du pin ou du sapin avant la vue de la planche. Supposons alors que l'on nous demande de prévoir la prochaine planche en prenant en compte seulement les probabilités a priori. Si une décision doit être prise alors il serait raisonnable de suivre la règle suivante : décider ω_1 si $P(\omega_1) > P(\omega_2)$; sinon choisir ω_2.

Ce processus paraît étrange puisque l'on prend toujours la même décision même si l'on connaît que les deux types d'arbres peuvent apparaître. L'efficacité de ce processus va dépendre fortement des probabilités a priori ; le processus apportera de meilleurs résultats si les probabilités sont très différentes, le pire des cas sera obtenu quand les probabilités seront égales. Ainsi la probabilité de l'erreur sera égale à la plus petite des probabilités a priori. Dans la plupart des circonstances, on ne nous demande pas de prendre de décision avec aussi peu d'informations. Dans notre cas, supposons que l'on arrive à distinguer que la planche de pin est plus claire que celle du sapin alors cette mesure de luminosité x devient un critère important dans notre décision. Cependant les différentes planches auront des couleurs diverses ; de ce fait, on exprime cette variable dans des termes probabilistes. On considère alors x comme une variable aléatoire continue qui dépend de l'état de la nature. Soit $p(x|\omega_j)$ la fonction de la densité de probabilité conditionnelle à l'état de nature de x, c'est-à-dire la fonction de la densité de probabilité de x sachant l'état de nature ω_j. Ainsi la différence entre $p(x|\omega_1)$ et $p(x|\omega_2)$ décrit la différence entre le teint d'une planche de pin et d'une planche de sapin.

Supposons à présent que nous connaissons les probabilités a priori $P(\omega_j)$ ainsi que les densités conditionnelles $p(x|\omega_j)$. Supposons également que nous pouvons mesurer le teint d'une planche et trouver la valeur de x. Comment cette mesure peut influencer

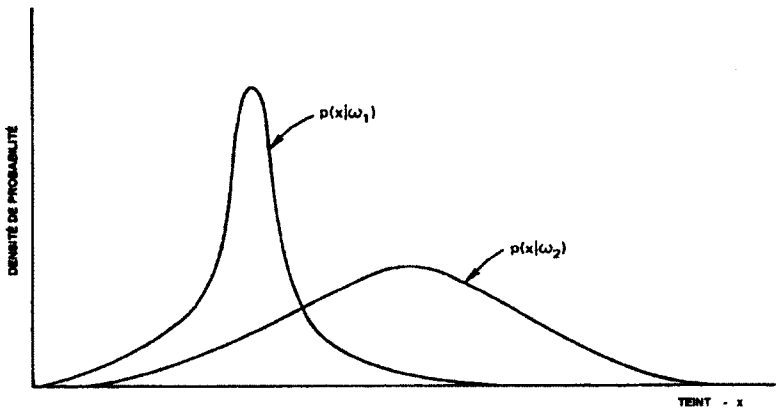

figure 2.6 – Exemples de fonctions de densité de probabilités conditionnelles

notre décision concernant l'état de nature ? La réponse à cette question est la loi de Bayes :

$$P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)}$$

avec

$$p(x) = \sum_{j=1}^{2} p(x|\omega_j)P(\omega_j)$$

La loi de Bayes montre comment l'observation de x permet d'exprimer la probabilité a posteriori $P(\omega_j|x)$ à partir de la probabilité a priori $P(\omega_j)$.

Ainsi, le théorème de Bayes définit la probabilité conditionnelle ou "la probabilité a posteriori" d'une hypothèse B (c'est-à-dire sa probabilité après l'apparition de l'évidence A) à l'aide de "la probabilité a priori" de A, "la probabilité a priori" de B et de la probabilité conditionnelle de A sachant B. Le théorème est valide dans tous les domaines de probabilités, et il est souvent utilisé en sciences.
Ce qui donne la formule suivante en prenant deux événements A et B et en ayant $P(A) \neq 0$,

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)}$$

Avec :

- P(A) la probabilité a priori (ou "inconditionnelle" ou "marginale") de A ; dans le sens qu'elle ne prend en compte aucune information concernant B.
- P(A|B) est la probabilité conditionnelle de A sachant B. Elle est aussi appelée la probabilité a posteriori puisqu'elle dépend de l'événement B.
- P(B|A) est la probabilité conditionnelle de B sachant A. Elle est aussi appelée la vraisemblance.
- P(B) est la probabilité a priori ou marginale de B.

Supposons qu'on aie plus d'un attribut à considérer, on va utiliser l'algorithme de Bayes naïf qui est basé sur les probabilités conditionnelles. L'algorithme se base sur le théorème de Bayes pour trouver la probabilité d'apparition d'un événement en fonction d'un événement qui est déjà apparu. Il est approprié pour la classification de données de grande dimension. Il faut faire l'hypothèse naïve que les attributs qui décrivent la donnée soient conditionnellement indépendants. Cet algorithme classifie les données en deux étapes :

1. étape d'apprentissage : en utilisant un échantillon d'apprentissage, l'algorithme estime les paramètres de la distribution de probabilité, en supposant que les attributs soient conditionnellement indépendants étant donné la classe.
2. étape de prédiction : pour chacune des données de test, l'algorithme calcule la probabilité a posteriori que cette donnée appartienne à chacune des classes. L'algorithme classifie alors cette donnée selon la plus grande probabilité a posteriori.

L'hypothèse de l'indépendance conditionnelle des classes simplifie grandement l'étape d'apprentissage puisqu'elle permet d'estimer la densité des classes conditionnelles unidimensionnelles pour chacun des attributs pris individuellement.

2.3.2 Classification bayésienne naïve

Paul Graham constate qu'en effet la faiblesse des spammeurs est le spam lui-même puisqu'il doivent à tout prix envoyer leur courriel. il constate que jusqu'alors toutes les techniques mises en place se font contourner. En effet, de son temps, l'approche statistique n'est pas utilisée pour détecter les spam. La plupart des programmeurs écrivent des programmes qui reconnaissent des caractéristiques bien précises des spams dont

ils sont victimes. Ils constatent, par exemple, que les spams qu'ils reçoivent, commencent toujours par "Dear Friend" ou ont un objet tout en majuscule et qui finit par 8 points d'exclamation. En effet, on peut filtrer le spam à l'aide de ce genre de caractéristiques très aisément. Le filtre va fonctionner au début et il faudra au fur et à mesure mettre à jour les règles de détection qui seront de plus en plus nombreuses et de plus en plus spécifiques pour conserver une efficacité décente. On aura alors des résultats qui tourneront aux alentours de 75% de spams détectés et moins de 5% de faux positifs. Si on utilise des règles beaucoup plus strictes, cela entraînera plus de faux positifs ce qui est très dommageable dans la plupart des cas. Effectivement, plus le filtre antispam sera performant pour bloquer le spam, d'une part, l'utilisateur va être plus confiant et va moins souvent vérifier son répertoire de spam ; d'autre part, les seules fois où il va vérifier ce dernier, ses courriers légitimes vont être noyés dans la masse de spams détectés.

La lutte antispam était alors une course entre les spammeurs et les filtres ; cela impliquait un travail constant de développement. Paul Graham part ainsi du constat qu'il faut une approche évolutive, fiable et facile à mettre en œuvre. Il utilise ainsi les filtres bayésiens. Nous allons donc nous concentrer sur ces filtres bayésiens naïfs.

Nous allons décrire ici le classificateur bayésien. Nous représentons chaque message par un vecteur $\vec{x} = \langle x_1, x_2, x_3, ..., x_n \rangle$, avec $x_1, ..., x_n$ sont les valeurs des attributs $X_1, ..., X_n$. Selon Sahami et al. [38], on utilise des attributs binaires : $X_i = 1$ si la caractéristique représentait par X_i est présente dans le courriel, sinon $X_i = 0$. Dans notre cas, les attributs correspondent aux mots, c'est-à-dire chaque attribut présente la présence d'un mot. Pour sélectionner à partir de tous les attributs, on calcule l'information mutuelle (MI) de Sahami et al. de chaque attribut X avec la variable catégorique C (spam ou légitime) :

$$MI(X;C) = \sum_{x \in \{0,1\}\ ,\ c \in \{spam, légitime\}} P(X = x, C = c).log\frac{P(X = x, C = c)}{P(X = x).P(C = c)}$$

Les attributs avec les MI les plus élevés sont sélectionnés. Grâce au théorème de Bayes et le théorème des probabilités totales [38], soit le vecteur $\vec{x} = \langle x_1, ..., x_n \rangle$ d'un document d, la probabilité que d appartienne à la catégorie c est :

$$P(C = c|\vec{X} = \vec{x}) = \frac{P(C = c) \cdot P(\vec{X} = \vec{x}|C = c)}{\sum_{k \in \{spam, légitime\}} P(C = k) \cdot P(\vec{X} = \vec{x}|C = k)}$$

Les probabilités $P(\vec{X}|C)$ sont quasiment impossibles à estimer directement (les valeurs possibles de $\vec{X}$ sont multiples, et sont un problème d'éparpillement des données). Le classificateur bayésien naïf fait l'hypothèse simplificatrice que $X_1, ..., X_n$ soient conditionnellement indépendants par rapport à la catégorie C. Alors :

$$P(C = c|\vec{X} = \vec{x}) = \frac{P(C = c) \cdot \prod_{i=1}^{n} P(X_i = x_i|C = c)}{\sum_{k \in \{spam, légitime\}} P(C = k) \cdot \prod_{i=1}^{n} P(X_i = x_i|C = k)}$$

où $P(X_i|C)$ et $P(C)$ peuvent être facilement estimées comme étant des fréquences relatives du corpus d'entraînement (c'est-à-dire pour $P(X_i|C)$ le nombre de courriels de la catégorie C où X_i est présent divisé par le nombre de courriels de la catégorie C).

Comme on l'a vu précédemment, le fait de classifier un courriel légitime en spam est plus préjudiciable que de laisser passer un spam à travers le filtre. On va noter $L \rightarrow S$ le fait de classifier des courriels légitimes en spam et $S \rightarrow L$ le fait de classifier des spams en tant que courrier légitime. On considère que $L \rightarrow S$ est λ fois plus coûtant que $S \rightarrow L$, on classifie alors un message comme étant un spam si :

$$\frac{P(C = spam|\vec{X} = \vec{x})}{P(C = légitime|\vec{X} = \vec{x})} > \lambda$$

Dans la mesure où l'hypothèse d'indépendance et que les probabilités estimées soient précises, un classificateur respectant ce critère apporte de bons résultats [8]. Dans ce cas-ci, $P(C = spam)|\vec{X} = \vec{x}) = 1 - P(C = légitime|\vec{X} = \vec{x})$, ce qui nous donne une autre expression du critère :

$$P(C = spam|\vec{X} = \vec{x}) > t, \text{ avec } t = \frac{\lambda}{1 + \lambda}, \lambda = \frac{t}{1 - t}$$

À titre de référence, Sahami et al. fixent le seuil t à 0.999 (donc $\lambda = 999$) ; ce qui signifie que bloquer un courriel légitime est comparable à laisser passer 999 spams à travers le filtre. On choisit un seuil assez important surtout dans les cas où l'on supprime tout de suite un courriel considéré comme étant un spam ; dans ce cas,

classifier un courrier légitime comme étant un spam signifierait perdre ce courriel. Cependant, aujourd'hui on utilise souvent des valeurs de λ moins importantes et on ne supprime pas tout de suite le spam. Le spam est souvent mis dans un répertoire pour une durée déterminée avant sa suppression définitive.

2.3.3 Modèle de représentation

Il existe deux manières différentes de représenter les caractéristiques dans un classificateur bayésien naïf. Ces deux méthodes vont donner des résultats différents au moment de la classification. Pour montrer la différence de ces deux modèles, nous allons utiliser un exemple d'un extrait de courriel que nous pourrions recevoir comme spam.

exemple : "Get rich fast!!! Visit now our online..."

Nous allons également utiliser les notations suivantes dans ce qui suit pour alléger l'écriture : c_i une classe et d un courriel. Ainsi on pourra noter le théorème de Bayes comme suit :

$$p(c_i|d) = \frac{p(c_i)p(d|c_i)}{p(d)}$$

avec $p(d) = \sum_{j=1}^{k} p(c_j)p(d|c_j)$ et k le nombre de classes.

Selon McCallum et Nigam [28] qui ont réalisé des expériences sur 5 corpus différents, le modèle de Bernoulli multivarié donne de bons résultats avec des vocabulaires de petites dimensions, mais le modèle multinomial donne de meilleurs résultats avec des vocabulaires de dimensions plus grandes.

Modèle de Bernoulli multivarié

Dans le modèle de Bernoulli multivarié, le document d est un vecteur binaire sur l'espace des mots. Soit un vocabulaire V, chaque attribut x_i est égal à 0 ou 1 selon

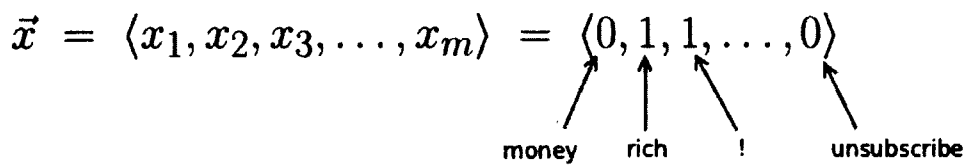

figure 2.7 – Le vecteur $\vec{x}$ dans un modèle de Bernoulli multivarié

que le token t_i issu de ce dictionnaire est présent ou non dans le courriel. La figure 2.7 illustre cet exemple.

Avec une telle représentation, nous faisons l'hypothèse de Bayes naïf que la probabilité d'apparition de chaque mot dans un document est indépendante de l'occurrence des autres mots. Ainsi, la probabilité d'un mot étant donné sa classe est le produit des probabilités de chaque mot de ce document selon la classe donnée.

$$p(d|c_i) = \prod_{j=1}^{m} p(x_j|c_i)$$

On peut donc voir un document comme étant une suite de plusieurs expériences de Bernoulli, une pour chaque mot du vocabulaire V. On peut remarquer également que ce modèle ne considère pas la fréquence d'apparition des mots dans un document.

Modèle multinomial

Dans le modèle multinomial, au contraire, on considère la fréquence d'apparition des mots dans un document. Ainsi, chaque attribut x_i du vecteur $\vec{x}$ représente le nombre d'occurrences dans le message de chaque token t_i du vocabulaire V. La figure 2.8 illustre cet exemple. N_j le nombre de mots du courriel va être égal à $\sum_j x_j$.

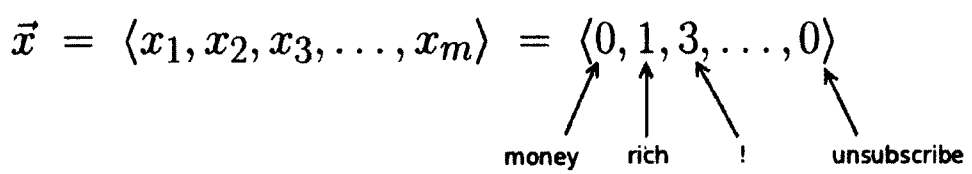

figure 2.8 – Le vecteur $\vec{x}$ dans un modèle multinomial

Dans ce modèle, un document est une séquence ordonnée de mots construits à partir du vocabulaire V. On suppose que la longueur du document ne dépend pas de

sa classe et on fait l'hypothèse de Bayes naïf que la probabilité d'apparition d'un mot dans un document est indépendante de sa position dans le document et des autres mots du document. On peut donc calculer la probabilité d'un mot selon la classe, avec la formule suivante :

$$p(d|c_i) = p(|d|)d! \prod_{j=1}^{m} \frac{p(x_j|c_i)^{N_j}}{N_j!}$$

2.4 Utilisation de filtre bayésien naïf pour amélio-rer la classication

Dans cette sous section, nous allons voir des approches utilisant les filtres bayésiens qui améliorent les résultats sur la détection de courriers indésirables et de courriers légitimes.

2.4.1 Sparse Binary Polynomial Hashing (SBPH) et CRM114

Sparse Binary Polynomial Hashing (SBPH) [47] est un moyen de créer un grand nombre de caractéristiques distinctifs à partir d'un texte entrant. Le but est de créer un grand nombre de caractéristiques parmi lesquelles plusieurs seront invariantes sur un large corpus de spam (ou non spam).

Ce processus se déroule en suivant ces étapes :

1. Faire glisser une fenêtre de N mots sur tout le texte entrant.
2. Pour chaque position de la fenêtre, générer un ensemble de sous-phrases qui conserve l'ordre des mots et qui est une combinaison des mots de la fenêtre.
3. Calculer le hashcode 32-bits de ces sous-phrases.

Ainsi, une caractéristique SBPH peut être :
- Hashcode d'un mot simple ("S1618")
- Hashcode d'une sous-phrase ("pour acheter")
- Hashcode de sous-phrases avec des trous ("pour <acheter> du viagra", "pour <> du viagra")

Chaque mot d'un texte va affecter $2N - 1$ caractéristiques, on utilise $N = 5$ (une fenêtre glissante de 5 mots) dans CRM114. De ce fait, il y aura un nombre beaucoup plus important de caractéristiques que de mots dans le texte original.

Exemple de SBPH :

Étape 1 : on fait glisser une fenêtre de N (5) mots sur le texte entrant.

You can Click here to buy viagra online NOW

Ce qui donne :
You can Click here to buy viagra online NOW
You can Click here to buy viagra online NOW
You can Click here to buy viagra online NOW
You can Click here to buy viagra online NOW
You can Click here to buy viagra online NOW

Étape 2 : on génère des sous-phrases à partir des mots de chaque fenêtre.

A partir de "Click here to buy viagra" on obtiendra :

Click				
Click	here			
Click		to		
Click	here	to		
Click			buy	
Click	here		buy	
Click		to	buy	
Click	here	to	buy	
Click				viagra
Click	here			viagra
Click		to		viagra
Click			buy	viagra
Click	here		buy	viagra
Click		to	buy	viagra
Click	here	to		viagra
Click	here	to	buy	viagra

Noter le comptage binaire utilisé, c'est pour cela que l'on parle de "Binary" dans SBPH.

Étape 3 : obtenir le hashcode 32 bits des caractéristiques obtenus.

On évalue ensuite ces caractéristiques grâce à un classificateur bayésien naïf. Ce classificateur agit en 2 étapes :

- Apprentissage : chaque caractéristique est marquée parmi les millions de caractéristiques d'un des deux fichiers de caractéristiques (spam ou non spam).
- Classification : le score de chaque caractéristique que l'on obtient dans les deux fichiers permet d'obtenir une estimation du caractère du courriel (spam ou non spam).

$$P(F|C) = 0.5 + \frac{(F_C - F_{\bar{C}})}{2 * MaxF}$$

avec F une caractéristique, C une classe (spam ou non spam), $\bar{C}$ le complémentaire de C et MaxF la valeur max de F.

$$P(C|F) = \frac{P(F|C) * P(C)}{P(F|C) * P(C) + P(F|\bar{C}) * P(\bar{C})}$$

Le langage CRM114 a été créé spécialement pour créer des filtres génériques. Le CRM114 prend seulement des chaînes de caractères et il ne supporte pas automatiquement les données de type numérique ou des données structurées. Les opérations de bases du CRM114 ont pour but de réaliser un couplage entre expressions régulières et chaînes de caractères.

La combinaison SBPH/BCR obtient de très bons résultats mais est confrontée aux mutations des spams. Certaines études approximent, par simples observations, l'apparition de nouveaux spams grâce à la formule suivante : $NouveauxSpams = TotalSpams^{0.001*jours}$

2.4.2 Filtre utilisant les Machines à Vecteurs de Support ou Séparateur à Vaste Marge (SVM)

Machines à Vecteurs de Support (Support Vecteur Machine, SVM)

Un séparateur à vaste marge (SVM) est une méthode de classification binaire par apprentissage supervisé, elle a été introduite par Vapnik en 1995. Cette technique se base sur l'existence d'un classificateur linéaire dans un espace approprié. Étant donné un problème de classification à deux classes, elle fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonctions dites noyaux (kernel) qui permettent une séparation optimale des données. Le but des SVM est de trouver un hyperplan (H) qui va séparer deux ensembles de points. De plus, les deux points les plus proches de cet hyperplan sont appelés vecteurs de support (voir figure 2.9).

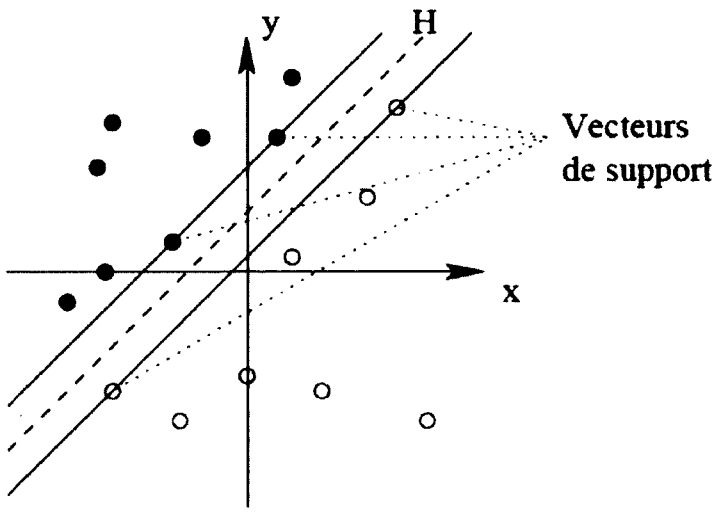

figure 2.9 – Hyperplan et vecteurs de support

Il existe plusieurs hyperplans valides mais grâce aux SVM on obtient l'hyperplan optimal. Formellement, cela revient à chercher un hyperplan dont la distance minimale à l'ensemble d'apprentissage est maximale. On appelle cette distance *marge* entre l'hyperplan et les exemples. L'hyperplan séparateur optimal est celui qui maximise cette marge, d'où l'appellation séparateurs à vaste marge. En effet, le fait d'avoir une marge plus grande procure plus de sécurité lorsque l'on classe un nouvel élément. Dans la figure 2.10, la partie droite illustre le fait qu'avec un hyperplan optimal, un nouvel élément reste bien classé alors qu'il tombe dans la marge et au contraire dans la partie de gauche, on constate qu'avec une plus petite marge, le nouvel élément est mal classé.

Parmi les modèles de SVM, on constate les cas linéairement séparables (comme sur les figures 2.9 et 2.10) et les cas non linéairement séparables. Dans la plupart des problèmes, il n'y a pas de séparation linéaire possible entre les données, et de ce fait, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables.

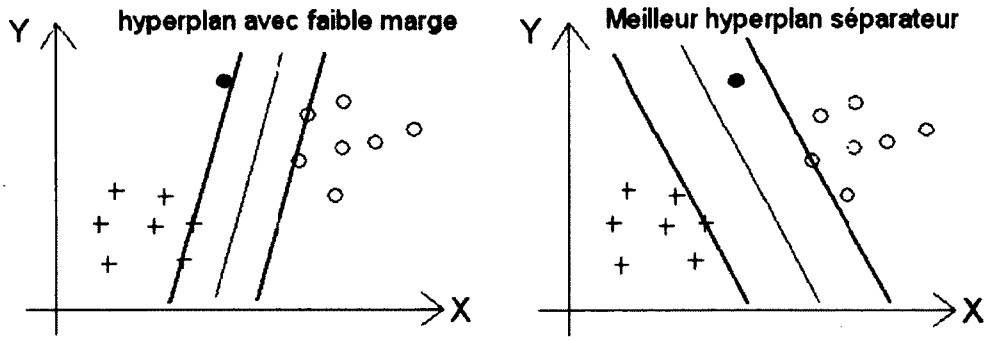

figure 2.10 – L'hyperplan qui donne une plus grande marge

Pour les cas non linéairement séparables, on change l'espace des données. La transformation non linéaire des données peut permettre une séparation linéaire des exemples dans un nouvel espace grâce à un changement d'espace de plus grande dimension. Cette nouvelle dimension est appelée *espace de redescription.* Cette transformation non linéaire est réalisée via une fonction *noyau.* En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur des SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application.

L'utilisation des SVM pour classifier le spam

Une des utilisations possibles pour effectuer la classification des spams est de combiner le SVM avec un classificateur bayésien naïf. Nous allons étudier cette possibilité à travers la proposition de Chui-Yu Chiu et Yuan-Ting Huan [5]. Ils partent du constat que d'une part, les classificateurs bayésiens naïfs obtiennent de bons résultats dans la détection des spams et d'autre part que les SVM donnent de bons résultats dans le text-mining.

Sélection des caractéristiques

Pour réaliser la sélection des caractéristiques, Chiu et Huan choisissent d'utiliser TF-IDF. Cette méthode est largement utilisée dans le domaine du text-mining. Elle permet d'évaluer l'importance d'un mot contenu dans un document, relativement à un corpus donné. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans le corpus.
La fréquence (tf) d'un mot ici va correspondre au nombre d'occurrences de ce mot dans le texte. On définit également la fréquence inverse de document (idf) comme étant la mesure de l'importance du mot dans l'ensemble du corpus. Elle se calcule selon la formule suivante :

$$idf_t = \log \frac{|D|}{|d_t : t \in d|}$$

avec $|D|$ le nombre total de documents du corpus, et $|d_t : t \in d|$ le nombre de documents où le terme t apparaît.

Ainsi on obtient, le poids d'un mot dans le corpus par le produit des deux mesures :

$$tfidf_t = tf_t * idf_t$$

La combinaison du classificateur bayésien naïf et du SVM

Généralement, on utilise le résultat précédent comme entrée du SVM mais dans le domaine de la détection de spams, ce procédé peut être insuffisant. Ainsi, Chiu et Huan calculent les valeurs d'entrées du SVM en faisant la multiplication entre les résultats de TF-IDF et des probabilités conditionnelles obtenues grâce au classificateur bayésien naïf.

Les différentes étapes de la figure 2.11 sont les suivantes :

1. Sélectionner les caractéristiques en utilisant TF-IDF

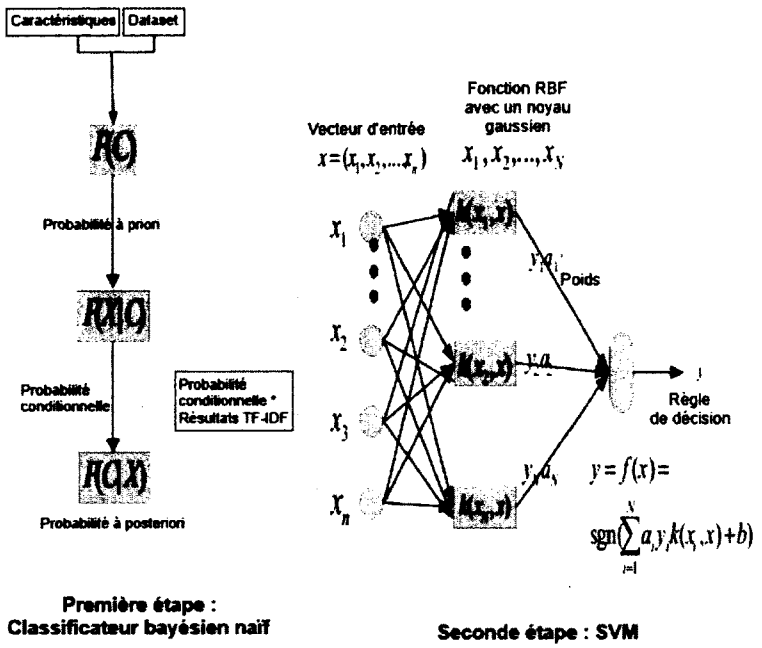

figure 2.11 – La combinaison d'un classificateur bayésien naïf et d'un SVM

2. Calculer les probabilités a priori et conditionnelles du classificateur bayésien naïf.
3. Calculer les probabilités a posteriori du classificateur bayésien naïf et classifier l'ensemble.
4. Entraîner le SVM en combinant les probabilités conditionnelles du classificateur bayésien naïf et les résultats de TF-IDF.
5. Classifier à l'aide du classificateur SVM en combinant les probabilités conditionnelles du classificateur bayésien naïf et les résultats de TF-IDF.

Nous avons, à travers ce chapitre, réalisé une revue de la classification des spams en utilisant un classificateur bayésien naïf. Ce classificateur est très utilisé dans la littérature et dans les solutions aujourd'hui disponibles pour réaliser la détection du spam. Ce classificateur bien qu'il donne des résultats acceptables est perfectible, d'où son utilisation combinée à un SVM ou SPBH. SPBH qui donne des résultats probants

a été une source d'inspiration pour notre méthode que nous allons présenter dans le chapitre suivant.

Chapitre 3

La classification de données catégorielles pour détecter les polluriels

À travers ce chapitre, je vais présenter l'approche que nous avons utilisée pour classifier les courriels. On va utiliser un algorithme qui sert à classifier des données catégorielles. Pour cela nous allons d'abord définir ce qu'est une donnée catégorielle avant de présenter notre algorithme et les outils que nous utilisons dans notre approche.

3.1 Données catégorielles et mesure de similarité

En statistique, une donnée catégorielle est une donnée d'un ensemble qui est constitué de variables catégorielles. Les variables catégorielles sont des variables évaluées sur une échelle nominale. C'est-à-dire une échelle qui propose des noms ou plus généralement des étiquettes. Les données catégorielles sont utilisées dans plusieurs domaines scientifiques et économiques comme des séquences biologiques, du texte, des relevés de transactions,etc. Ces données peuvent provenir d'observation tant qua-

litatives que quantitatives.

La mesure de similarité des données catégorielles est obtenue par la détection des dépendances chronologiques et des caractéristiques structurelles cachées au sein de ces données. Ainsi, un problème récurent dans la classification de données catégorielles est la détection de ces dépendances chronologiques et de ces caractéristiques structurelles cachées dans ces données.

On retrouve souvent la distance de Levenshtein [25], appelée aussi "Edit Distance" comme approche utilisée pour réaliser cette tache. Cette distance est calculée en obtenant le coût minimum pour transformer une séquence en une autre en utilisant des opérations d'insertion, suppression ou substitution (voir figure 3.1). L'alignement de séquences [33] est une autre approche communément utilisée pour trouver le meilleur couple entre deux séquences catégorielles en insérant des "gaps" (décalages) aux positions appropriées.

Cependant, ces deux méthodes ne fonctionnent pas pour des séquences qui comprennent des caractéristiques structurelles similaires dans un ordre chronologique différent puisqu'elles se basent sur la correspondance de motifs dans un ordre chronologique. De plus, ces deux approches utilisent une mesure de similarité qui dépend beaucoup du coût que l'utilisateur accorde à chacune des opérations d'insertion, suppression et substitution dans le cas de la distance de Levenshtein ou aux opérations de création de décalage et d'extension de décalage dans le cas d'alignement de séquences. La littérature fait aussi l'état de l'utilisation de l'approche N-Gram [20] qui est souvent couplée avec LSA (Latent Semantic Analysis) [42] que nous verrons plus loin.

Nous allons utiliser une mesure de similarité appelée SCS (Similarity measure for Categorical Sequences) présentée dans [17]

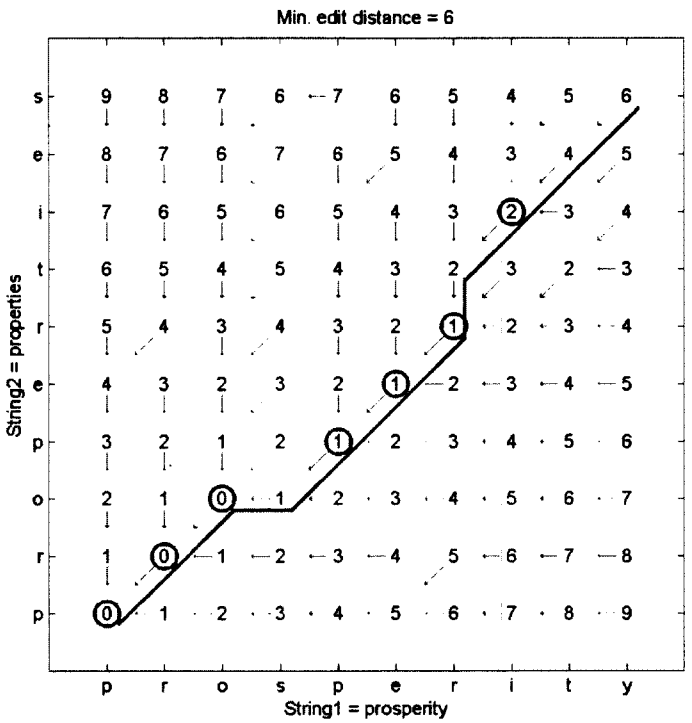

figure 3.1 – Exemple de calcul de distance de Levenshtein

3.2 Outils utilisés

3.2.1 *N*-grams

Dans plusieurs applications, il est nécessaire de quantifier de manière algorithmique la similarité entre deux chaînes composées d'un alphabet dont le nombre de symboles est défini. Le problème de mesure de similarité est un problème récurrent dans plusieurs domaines.

Un ***N*-gram** est une trame de N-caractères provenant d'une plus grande chaîne [20] [4]. Dans la littérature, ce terme peut aussi inclure la notion de co-apparition d'un ensemble de caractères dans une séquence (par exemple, un N-gram construit à partir du premier et troisième caractère d'une chaîne). Nous allons utiliser seulement

le premier type de N-gram.

N-gram collecte tous les grams possibles (c'est-à-dire motifs) d'une longueur fixée N. L'alphabet utilisé comprenant m caractères, on obtient m^N motifs possibles. Même si la valeur N fixée est un inconvénient majeur [30], la valeur de N est définie indépendamment de la structure intrinsèque des séquences, c'est-à-dire qu'elle ne dépend ni de l'alphabet ni de la longueur de la séquence. Selon la valeur de N, les résultats présenteront donc du bruit ou ne représenteront pas tous les motifs importants. De plus, tous les motifs de longueur N sont collectés sans considération qu'ils représentent du bruit ou un motif important ; ce qui augmente la probabilité de récolter du bruit.

L'intérêt de la mise en correspondance en utilisant les N-grams est que la présence d'une erreur ou d'une non correspondance sur un caractère n'affectera pas l'ensemble mais seulement une petite région puisque l'on ne compare plus des mots mais des séquences. Ainsi, on pourra trouver une similarité entre les séquences 'FREE VIAGRA' et 'FREE VI4GRA' si on utilise des 4-grams alors qu'une approche classique n'aurait pas détecté cette similarité. Finalement, en comptant les N-grams communs aux deux séquences, on peut obtenir une mesure de similarité qui est résistante aux erreurs ou modifications de texte.

3.2.2 Analyse Sémantique Latente (LSA)

L'analyse sémantique latente (LSA) [24] est une technique statistique automatisée qui permet d'extraire et d'induire des relations entre un ensemble de documents et de termes qu'ils contiennent. Ce n'est pas nécessairement un traitement du langage naturel traditionnel. Elle prend généralement en entrée du texte brut qui sera analysé comme des chaînes de caractères uniques et séparées en passages significatifs ou en échantillons comme des phrases ou paragraphes.

La première étape consiste à représenter le texte comme une matrice dans laquelle chaque ligne représente un mot unique, et chaque colonne représentera un passage de texte. Chaque cellule contiendra la fréquence de la présence du mot de la ligne

dans le passage donné par la colonne. Ensuite on applique une transformation à ces cellules dans laquelle chaque fréquence est pondérée par une fonction qui exprime à la fois l'importance de chacun des mots dans le passage lui-même ainsi que dans le corpus intégral. Par la suite, LSA applique une décomposition en valeur singulière (SVD) sur la matrice. Dans une SVD, une matrice rectangulaire X sera décomposée en un produit de trois nouvelles matrices :

- Une matrice P qui contient un ensemble de vecteurs de base orthonormés pour X, dits "d'entrée" ou "d'analyse" ;
- Une matrice W qui contient un ensemble de vecteurs de base orthonormés pour X, dits "de sortie" ;
- une matrice S qui contient les valeurs singulières de la matrice X.

Exemple :

Cet exemple utilise comme passages de texte les titres de neuf notes techniques, cinq sur l'interaction homme-machine (IHM), et quatre sur la théorie des graphes mathématiques, des sujets qui sont conceptuellement non disjoints. Ainsi, la matrice originale est composée de neuf colonnes, et nous lui avons donné douze lignes, chacune correspondant à un mot du contenu utilisé dans au moins deux des titres. Dans la figure 3.2, nous pouvons voir les trois étapes de l'analyse sémantique latente. À la première étape on compte la présence de chacun des mots (humans, interface, etc.) dans les 9 titres. Vu que chaque mot est présent au moins 2 fois la somme de chaque ligne est supérieure à 2. À l'étape 2, on réalise une SVD. Enfin l'étape 3 présente une reconstruction basée sur deux dimensions qui se rapproche de la matrice d'origine. Celle-ci utilise seulement les deux premières colonnes en gris des trois matrices de l'étape 2.

3.3 Motifs significatifs

Notre approche de classification ne se basant pas sur un traitement des courriels mot à mot mais plutôt en traitant ces données en considérant des motifs significatifs et une mesure de similarité particulière SCS (Similarity measure of Categorical Sequences). Nous allons décrire la méthode que nous utilisons pour trouver ces motifs.

Étape 1: $\{X\} =$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Étape 2: SVD

$\{X\} = \{W\}\{S\}\{P\}$

$\{W\} =$

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

$\{S\} =$

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

$\{P\} =$

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

Étape 3:

$\{\hat{X}\} =$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.11	0.31	0.41	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

figure 3.2 – Exemple de LSA

3.3.1 Choix des motifs significatifs

Un motif significatif est obtenu grâce à la comparaison de deux séquences. On émet l'hypothèse que l'apparition de motifs significatifs dans un ordre non-chronologique (c'est-à-dire des motifs qui contiennent des caractéristiques structurelles similaires dans un ordre chronologique différent) est plus susceptible de survenir comme un phénomène local plutôt que global. Cette hypothèse est soutenue par l'observation de cette caractéristique dans des données réelles de diverses types.

Soit C un ensemble de séquences catégorielles, duquel X et Y sont une paire de séquences. Soit x et y une paire de sous-séquences appartenant respectivement à X et Y (x et y sont des variables représentant n'importe quelles sous-séquences de X et Y). On construit un ensemble d'appariement (matching set) $E_{X,Y}$ en collectant toutes les paires possibles de sous-séquences x et y qui satisfassent les conditions suivantes :

$$E_{X,Y} = \left\{ x, y \;\middle|\; \begin{array}{c} |x| = |y| \\ |x \cap y| > N_{X,Y} \\ |x \setminus y| < N_{X,Y} \\ \forall x', y' \in E_{X,Y} \Rightarrow (x \not\subset x') \vee (y \not\subset y') \end{array} \right\}$$

Les symboles x' et y' représentent des variables au même titre que x et y, $N_{X,Y}$ représente à la fois le nombre minimum de positions appariées avec des symboles similaires entre x et y, et aussi le nombre maximum toléré de positions appariées avec des symboles différents. Nous allons voir dans la section suivante comment obtenir ce nombre $N_{X,Y}$.

Ainsi, $|x| = |y|$ signifie que x et y sont de même longueur, $|x \cap y| > N_{X,Y}$ signifie que x et y contiennent plus de $N_{X,Y}$ positions avec des symboles similaires, $|x \setminus y| < N_{X,Y}$ signifie que x et y contiennent moins de $N_{X,Y}$ positions avec des symboles différents et $\forall x', y' \in E_{X,Y} \Rightarrow (x \not\subset x') \vee (y \not\subset y')$ signifie que pour toute paire de sous-séquences x', y' appariées de l'ensemble $E_{X,Y}$ au moins un de x ou y n'est pas inclus respectivement dans x' ou y', aussi bien en terme de sa composition qu'en terme de leur position dans leur séquence. De ce fait, $E_{X,Y}$ permet de capturer les paires de motifs x, y qui ont

une similarité chronologique même s'ils sont dans des ordres non chronologiques selon leur position dans les séquences X et Y.

3.3.2 Longueur des motifs significatifs

L'objectif est de détecter et utiliser ces motifs significatifs qui représentent le plus fidèlement la structure de la séquence, tout en minimisant l'influence des motifs qui apparaissent par chance ou qui représentent du bruit. Ainsi, la mesure de similarité va utiliser les plus longs motifs significatifs (c'est-à-dire les occurrences multiples) dans l'appariement puisque plus le motif est long, moins il y a de chances que le motif représente du bruit. Pour chaque paire de séquence X et Y, la mesure utilise la théorie développée par Karlin et al. [15] pour calculer la longueur minimale des motifs significatifs appariés qui sera la valeur assignée à $N_{X,Y}$.

Selon Karlin et al. [15], la longueur attendue $K_{X,Y}$ du plus long motif communs présent au moins 2 fois par chance dans 2 m-catégories de séquences X et Y est obtenu comme suit :

$$K_{X,Y} = \frac{log(|X|^2 + |Y|^2) + log\lambda_{X,Y}(1 - \lambda_{X,Y}) + 0,57}{-log\lambda_{X,Y}}$$

$$\lambda_{X,Y} = max\left(\sum_{i=1}^{m}(p_i^X)^2, \sum_{i=1}^{m}(p_i^Y)^2\right)$$

$$\sigma_{X,Y} \approx \frac{1,28}{|log\lambda_{X,Y}|}$$

où p_i^X et p_i^Y sont généralement les fréquences de la $i^{ème}$ catégorie observée respectivement dans X et Y et $\sigma_{X,Y}$ est la déviation asymptotique standard de $K_{X,Y}$.

Nous utilisons le critère de conservation proposée par Karlin *et al.* [15] qui dit que pour une paire de séquences X et Y, un motif observé deux fois est désigné comme statistiquement significatif s'il a une longueur supérieure à $K_{X,Y}$ d'au moins 2 déviations standards. Ainsi, en construisant l'ensemble d'appariement $E_{X,Y}$, on extrait tous les motifs respectant ce critère. Ce qui veut dire que pour une paire de séquence

X et Y, on calcule une valeur spécifique $N_{X,Y} = K_{X,Y} + 2\sigma_{X,Y}$. Selon Karlin *et al.* [15], ce critère garantit qu'un motif apparié désigné comme statistiquement significatif (c'est-à-dire un motif qui s'apparie en terme de structure de séquence) a moins d'une chance sur cent d'être présent par hasard.

3.3.3 La matrice motif-séquence

Soit C un ensemble de données catégorielles, dans lequel X et Y sont deux séquences différentes, pour lesquels $N_{X,Y}$ est la longueur minimale pour les motifs significatifs. Soit E l'ensemble de tous les ensembles d'appariement :

$$E = \bigcup_{X,Y \subset C} E_{X,Y}$$

et

$$N_{min} = \min_{X,Y \subset C} N_{X,Y}$$

Pour obtenir la matrice motif-séquence T, on collecte tous les N_{min}-grams de tous les motifs significatifs de l'ensemble E. Ainsi, pour un ensemble de séquences construit de m catégories on pourrait obtenir un maximum de $m^{N_{min}}$ N_{min}-grams possibles. Soit E_X le sous ensemble de tous les ensembles d'appariement en rapport avec la séquence X :

$$E_X = \bigcup_{Y \subset C} E_{X,Y}$$

La valeur (initialisée à zéro) de $T_{i,X}$ représentant l'intersection de la $i^{ème}$ ligne et de la colonne correspondante à la séquence X, est simplement augmentée par les occurrences du $i^{ème}$ N_{min}-grams collectés et appartenant au sous-ensemble E_X. Après avoir construit la matrice T, on supprime les lignes correspondantes aux N_{min}-grams qui existent au plus dans une séquence. Selon [17], le nombre de lignes W est plus petit que $m^{N_{min}}$.

L'avantage de cette approche est que chaque séquence dans l'ensemble des séquences contribue à la capture de caractéristiques structurelles et des dépendances

chronologiques de toutes les autres séquences de l'ensemble. Plus la fréquence d'apparition d'un motif dans les séquences est grande, plus il sera représenté dans la matrice motif-séquence T. De plus, la matrice T est construite en utilisant seulement les N_{min}-grams correspondants aux motifs significatifs collectés.

3.4 CLASS

CLASS est une méthode qui permet de réaliser la classification de données catégorielles en utilisant les relations structurelles obtenues grâce aux informations globales tirées d'un grand nombre de séquences au lieu d'une simple comparaison par paire des séquences. CLASS ne dépend pas de l'ordre chronologique ou non-chronologique de la structure des données. Ainsi, cette approche permet de travailler sur des données catégorielles qui peuvent contenir des motifs significatifs à des positions non chronologiquement équivalente. Cette méthode procède suivant ces deux étapes :

- Étape 1 : Apprentissage
    1. Extraction des motifs significatifs de toutes les séquences de l'ensemble d'apprentissage.
    2. Application de l'approche N-Gram sur les motifs significatifs pour construire une matrice motif-séquence dans laquelle les lignes correspondent aux motifs significatifs et les colonnes représentent les séquences des données d'apprentissage.
    3. On effectue une SVD (Singular Value Decomposition) sur cette matrice motif-séquence pour obtenir une représentation de l'espace vectoriel des séquences de l'ensemble d'apprentissage.
- Étape 2 : Test
    1. Extraction des motifs significatifs de toutes les séquences de l'ensemble de test.
    2. Application de l'approche N-Gram sur les motifs significatifs pour construire une matrice motif-séquence semblable à celle construite durant l'étape d'apprentissage.

3. On projette cette matrice sur le même espace vectoriel que celui obtenu pendant la phase d'apprentissage.
4. On applique SNN pour prédire la classe de chacune des séquences de l'ensemble de test.

3.4.1 Idée principale

Pour le traitement de corpus de texte en langage naturel, on utilise fréquemment l'Analyse Sémantique Latente (LSA) pour extraire des relations cachées entre des documents. Cela se fait grâce à une matrice de mot-document.

L'Indexation Sémantique Latente (LSI) [2] [7] est une approche qui permet de surmonter le problème d'appariement lexical en utilisant des indices conceptuels calculés de manière statistique au lieu de rechercher des mots. Cette approche présume qu'il existe une structure latente ou cachée dans l'utilisation des mots puisqu'il y a une variabilité dans le choix des mots. Elle permet ainsi de construire une matrice $W * L$ de termes par document à partir de l'analyse des documents du corpus de texte. Cette matrice dans laquelle W représente le nombre de mots du dictionnaire et L le nombre de documents permet de réaliser une représentation globale de l'information contenue dans les documents. Du fait, que l'on applique cette méthode généralement sur des ensembles de documents qui contiennent peu de mots, cette matrice est clairsemée mais elle permet tout de même d'extraire des relations qui sont difficiles à identifier.

Dans le domaine des données catégorielles, les motifs qui contiennent de l'information significative (discriminante) ne sont pas directement disponibles comme les mots dans le cas d'une analyse de texte d'un langage naturel. Ces motifs jugés importants doivent être extraits et les motifs non-significatifs eux doivent être rejetés. Comme pour une matrice mot-document, notre méthode utilise une matrice de motif-séquence pour détecter et extraire l'information contenue entre différentes séquences catégorielles. Ainsi dans notre cas, W représente le nombre de motifs significatifs possibles et L le nombre de séquences catégorielles. On utilise l'approche de Song et Park [42] pour détecter et collecter les motifs significatifs et ainsi construire la matrice motif-

séquence. Cette approche est basée sur les algorithmes génétiques.

Pour la suite, T^A représentera la matrice motif-séquence construite à partir de l'ensemble d'apprentissage A, et T^B représentera la matrice motif-séquence obtenue à partir de l'ensemble de test B.

3.4.2 La décomposition spectrale

En appliquant une décomposition spectrale sur la matrice motif-séquence, les séquences catégorielles sont projetées sur un espace vectoriel de dimensionnalité réduite. Ainsi, la mesure de similarité entre les différentes séquences est obtenue par la similarité cosinus entre les vecteurs. En effet, la similarité cosinus permet de calculer la similarité entre deux vecteurs à n dimensions en déterminant l'angle entre eux. Ainsi, soit deux vecteurs A et B, l'angle θ se calcule de la manière suivante :

$$\theta = arccos \frac{A.B}{\|A\|.\|B\|}$$

L'angle obtenu étant dans l'intervalle $[0, \pi]$, si le cosinus de l'angle obtenu est égal à -1 cela veut dire que les deux vecteurs sont exactement opposés, 1 veut dire qu'ils sont exactement similaires et 0 qui indique généralement l'indépendance. L'avantage d'appliquer LSA est que la mesure de similarité entre différentes séquences peut s'obtenir en utilisant l'information globale extraite de l'ensemble des séquences au lieu de comparer seulement une paire de séquences. Cet avantage est possible grâce à la décomposition spectrale qui transforme chaque séquence en un vecteur en utilisant tout l'ensemble des séquences, ce qui donne une portée globale à la mesure de similarité entre les différents vecteurs.

Représentation de l'ensemble de donnée d'apprentissage

Soit A l'ensemble d'entraînement, dans la matrice motif-séquence T^A, chaque séquence catégorielle est représentée par un vecteur colonne et les motifs sont représentés par des vecteurs de lignes. Dans cette représentation, les séquences catégorielles sont des points dans l'espace multidimensionnel engendrés par les motifs. Mais cette

représentation ne reflète pas les relations entre séquences ou motifs et les dimensions sont très grandes. On effectue une SVD sur la matrice T^A pour tirer avantage de cette représentation. La matrice T^A peut se décomposer en un produit de trois matrices, U une matrice unitaire gauche, Σ une matrice diagonale de valeur singulière positive ou nulle et V' la transposée de V un matrice unitaire droite :

$$T^A = U \times \Sigma \times V'$$

Les séquences catégorielles de A représentées par des vecteurs colonnes dans T^A sont projetées grâce à la décomposition spectrale dans un espace engendré par les vecteurs colonnes de la matrice V'. Soit $\hat{T}^A$ la représentation des colonnes vecteurs de T^A dans le nouvel espace tel que :

$$T^A = U \times \Sigma \times \hat{T}^A$$

Représentation de l'ensemble de donnée de tests

Selon la théorie de la décomposition spectrale [2], une séquence catégorielle j de l'ensemble d'apprentissage A représente comme un vecteur colonne $T^A_{.,j}$ dans la matrice T^A est projetée dans le nouvel espace en tant que $\hat{T}^A_{.,j}$ telle que :

$$T^A_{.,j} = U \times \Sigma \times \hat{T}^A_{.,j}$$

on obtient ainsi :

$$\hat{T}^A_{.,j} = \Sigma^{-1} \times U' \times T^A_{.,j}$$

Ce qui veut dire, qu'étant donné un vecteur requête q incluant les motifs significatifs collectés pour une séquence catégorielle Q, on peut représenter Q dans l'espace vectoriel avec $\hat{q}$ en effectuant la transformation suivante :

$$\hat{q} = \Sigma^{-1} \times U' \times q$$

Ainsi, la séquence catégorielle Q peut être comparée à toutes les autres séquences de A, en comparant le vecteur $\hat{q}$ à tous les vecteurs colonnes $\hat{T}^A$. Donc, une séquence

catégorielle j dans l'ensemble de test B représenté dans la matrice motif-séquence T^B par le vecteur colonne $T^B_{.,j}$ peut être transformé par un vecteur $\hat{T}^B_{.,j}$ dans l'espace vectoriel comme suit :

$$\hat{T}^B_{.,j} = \Sigma^{-1} \times U' \times T^B_{.,j}$$

Ou plus généralement pour toutes les séquences dans B :

$$\hat{T}^B = \Sigma^{-1} \times U' \times T^B$$

Pour voir les relations entre les séquences catégorielles X de A et Y de B, dans l'espace vectoriel en comparant les vecteurs colonnes $\hat{T}^A_{.,X}$ et $\hat{T}^B_{.,Y}$ avec la similarité cosinus.

3.4.3 L'algorithme SNN

SNN est inspiré de l'algorithme de classification KNN. L'algorithme KNN est une des méthodes de classification des plus utilisées et des plus simples qui est souvent utilisée quand on a pas beaucoup d'informations a priori sur la distribution des données. L'algorithme KNN classifie un objet en fonction des éléments déjà classifiés en cherchant les plus proches voisins en terme de caractéristiques [31]. Le plus grand problème de cette méthode est que la sensibilité des résultats dépend de la séléction du paramètre K. En particulier, quand pour une classe donnée, il existe moins de K voisins réels.

En effet, dans l'exemple de la figure 3.3, on veut classifier l'échantillon de test (représenté par l'étoile) par rapport à dix objets de deux classes distinctes (représentées par des triangles et des carrés), chacune des classes contient cinq objets. L'échantillon de test doit être classifié dans l'une des deux classes de manière exclusive.

En utilisant, KNN avec $K = 3$, l'échantillon de test est classifié dans la classe des triangles puisqu'il y a deux triangles pour seulement un carré dans le voisinage délimité par les plus petits tirets discontinus de l'échantillon de test. Par contre, si on choisit $K = 7$ alors l'échantillon sera classifié dans la seconde classe (représentée par

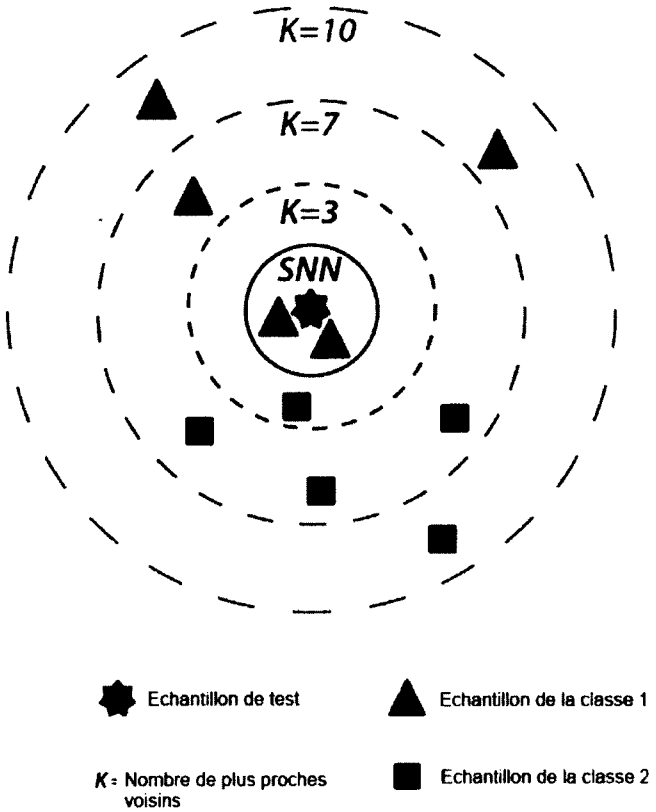

figure 3.3 – Exemple de différence entre SNN et KNN

des objets carrés) puisque le nombre d'objets représentés par des triangles est inférieur au nombre d'objets représentés par des carrés (voisinage contenu dans la zone à l'intérieur du cercle de tirets moyens). Enfin, si $K = 10$ alors on ne peut pas classifier de manière sûre l'échantillon puisque le nombre d'objets appartenant à chacune des classes triangle et carrée est égal.

Nous voyons à travers cet exemple, l'impact du choix de K, le nombre d'objets du voisinage, sur la qualité de la classification obtenue avec l'algorithme KNN. Ce choix affecte directement le taux de faux positifs ou de faux négatifs. Pour pallier à ces problèmes, on utilise SNN qui est un classificateur qui va ajuster de manière automatique la valeur de K dans KNN. En effet, l'avantage majeur de SNN sur KNN est le

fait qu'il permet de trouver la véritable valeur de K plutôt que d'utiliser une valeur fixée qui affecte la classification et peut ne pas représenter la vraie distribution des objets. Ainsi, SNN permet une classification plus précise surtout quand les voisinages doivent être de différentes tailles.

On peut voir dans la figure 3.3 que KNN a besoin d'une valeur fixée de K pour savoir quels objets sont dans le voisinage de l'échantillon de test alors que SNN est capable de distinguer quels objets sont réellement dans le voisinage de celui-ci. Dans l'exemple précédent, on voit que SNN considère seulement les deux objets de la classe triangle comme étant dans le voisinage de l'échantillon de test. Ce voisinage est représenté à l'intérieur du cercle continu. Cette classification est acceptable puisqu'en effet ces deux objets sont les plus proches de l'échantillon.
L'algorithme SNN utilise une méthode systématique basée sur le théorème de König-Huygens pour décider quelles séquences sont les plus similaires par rapport à une séquence donnée de l'ensemble des séquences.

Théorème 3.4.1 *En statistique et en théorie des probabilités, le théorème de König-Huygens est une identité remarquable reliant la variance et la moyenne.*
Pour toute variable aléatoire réelle X, on a :

$$Var(X) \equiv E[(X - E[X])^2] = E[X^2] - E[X]^2$$

Soit $\hat{T}^A$ la représentation de l'ensemble d'entraînement A dans l'espace généré et $\hat{T}^B$ la représentation de l'ensemble de test B dans le même espace. Soit $S_{X,Y}$ la mesure de similarité entre les séquences X et Y appartenant respectivement à l'ensemble A et B. Cette mesure est une similarité cosinus entre les vecteurs $\hat{T}^A_{.,X}$ et $\hat{T}^B_{.,Y}$. Soit R la séquence de B que l'on veut classifier.

Après avoir classé les séquences de A en fonction de leur similarité cosinus avec R de manière décroissante. On définit A_H et A_L un partitionnement de A en deux parties tel que A_H regroupe les séquences de A ayant une grande similarité avec R et

A_L regroupe les séquences de A ayant une faible similarité avec R. On a :

$$A_H \cup A_L = A$$

$$A_H \cap A_L = \emptyset$$

$$\forall i, j \in A \mid i \in A_H, j \in A_L \Rightarrow S_{i,R} > S_{j,R}$$

A_H et A_L sont des variables utilisées pour représenter tous les partitionnements possibles de A selon les conditions précédentes.

En utilisant le théorème de König-Huygens 3.4.1, l'inertie totale[1] I est calculée comme suit :

$$I = \sum_{i \in A_H} \left(S_{i,R} - \bar{S}_{A_H}\right)^2 + \sum_{j \in A_L} \left(S_{j,R} - \bar{S}_{A_L}\right)^2 + \left(\bar{S}_{A_H} - \bar{S}_{A_L}\right)^2$$

Avec $S_{i,R}$ et $S_{j,R}$ les similarités respectives entre i et R et entre j et R. $\hat{S}_{A_H}$ et $\hat{S}_{A_L}$ les moyennes de similarités respectives des partitions A_H et A_L (c'est-à-dire centres de gravité) telles que :

$$\hat{S}_{A_H} = \frac{1}{|A_H|} \sum_{i \in A_H} S_{i,R} \qquad et \qquad \hat{S}_{A_L} = \frac{1}{|A_L|} \sum_{j \in A_L} S_{j,R}$$

Le meilleur partitionnement de A est obtenu par le couple (A_H^*, A_L^*) qui maximise la valeur de l'inertie totale I. Ainsi, les séquences significatifs les plus similaires sont dans l'ensemble A_H^*.
Pour calculer $P(R|k)$, la probabilité pour qu'une séquence R appartienne à une classe k, nous utilisons la moyenne pondérée suivante :

$$P(R|k) = \frac{\sum_{i \in A_H^{*k}} S_{i,R}}{\sum_{i \in A_H^*} S_{i,R}}$$

avec ${A_H^*}^k$ le sous ensemble de A_H^* des séquences de classe k.
De ce fait, la séquence R va appartenir à la classe k qui maximise la valeur de $P(R|k)$.

1. L'inertie est la moyenne des carrées des distances des points au centre de gravité. L'inertie totale est égale à la somme de l'inertie interclasse et de l'inertie intraclasse.

3.5 Expérimentation et Comparaison

3.5.1 Ensembles de test

Pour expérimenter notre filtre, nous réalisons une panoplie de tests comparatifs. Pour ces tests, nous utiliserons deux ensembles de données connus dans le domaine de la classification du spam.

Enron dataset

L'ensemble de données Enron [18] a été rendu publique suite aux investigations sur la Enron corporation. Le corpus Enron brut contient 619 446 courriels provenant de 158 utilisateurs. Cet ensemble a été nettoyé en supprimant des répertoires comme les "discussion threads". Ces répertoires étaient présents chez la plupart des utilisateurs et n'étaient pas directement utilisés par ces utilisateurs mais étaient générés de manière automatique par le système. D'autres répertoires, comme le repértoire "all documents", contenaient aussi un grand nombre de messages dupliqués puisqu'ils étaient déjà présents dans d'autres répertoires.

Nous utilisons donc un ensemble de données nettoyé qui contient au final 200 399 courriels appartenant aux 158 utilisateurs avec une moyenne de 757 courriels par utilisateur. La figure 3.4 présente la distribution des courriels par utilisateur. Les utilisateurs sont ordonnés selon le nombre ascendant de messages qu'ils avaient sur l'axe des abscisses. Le nombre de messages est représenté dans une échelle logarithmique sur l'axe y. Le trait horizontal représente la moyenne des messages par utilisateur (757). Comme nous le voyons sur le graphe, les messages sont distribués de manière exponentielle, avec un petit nombre d'utilisateurs ayant un grand nombre de messages. Cependant, on remarque qu'il y a des utilisateurs ayant un nombre varié de messages de 1 à 100 000 messages, ce qui nous permet d'obtenir des données d'utilisateurs variés.

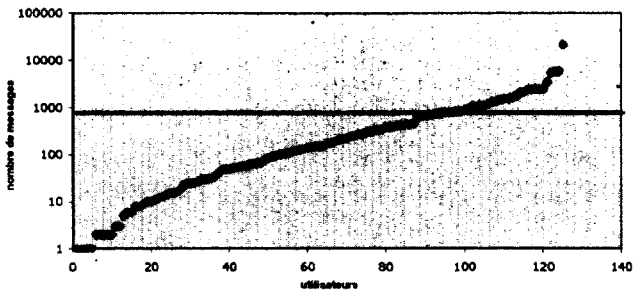

figure 3.4 – Distribution des messages par utilisateur

Ling-Spam dataset

L'ensemble de données Ling-Spam [39] est un mélange de courrier spam et légitime envoyés à travers la liste Linguist qui est une mailling liste modérée qui traite à propos de la science et la profession de la linguistique. Le corpus contient 2893 messages :

- 2412 messages légitimes, obtenu par téléchargement aléatoire d'une collection des archives de la liste et dont le texte ajouté par le serveur de la liste a été supprimé.
- 481 spams reçus par un des auteurs, les pièces-jointes, tag HTML, et doublons reçus le même jour ont été nettoyés.

Ainsi, les spams représentent approximativement 16% du corpus, un taux proche des taux rapportés par Cranor et LaMacchia [6], et Sahami et al. [38]. Cet ensemble n'est pas si spécifique et ne traite pas d'une thématique unique, il contient aussi bien des offres d'emploi ou des annonces sur de nouveaux logiciels par exemple.

L'ensemble des messages contenus dans Ling-Spam est petit en comparaison au benchmarks utilisés pour faire du classement de texte (c'est-à-dire texte mining) comme "Reuters corpus". Ling-Spam est partitionné en 10 parties ayant chacune la même proportion de spam. Ceci permet de réaliser des tests en utilisant 9 parties pour l'entraînement et 1 partie pour le test. Ainsi, nous allons utiliser cette particularité pour tester les effets de l'apprentissage sur notre approche.

Enron-Spam	CLASS utilisé avec					
	SPM et SNN	SPM et KNN (K=100)	SPM et KNN (K=10)	N-Gram et SNN	N-Gram et KNN (K=100)	N-Gram et KNN (K=10)
Enron 1	0.96	0.89	0.93	0.91	0.86	0.81
Enron 2	0.97	0.91	0.90	0.86	0.84	0.81
Enron 3	0.97	0.90	0.85	0.94	0.86	0.88
Enron 4	0.98	0.93	0.91	0.95	0.94	0.81
Enron 5	0.97	0.89	0.90	0.94	0.84	0.88
Enron 6	0.97	0.94	0.87	0.96	0.86	0.84
Moyenne	0.97	0.92	0.90	0.92	0.87	0.84

tableau 3.1 – CLASS sur les données "Enron"

3.5.2 Expérimentation

Pour évaluer notre approche, nous allons procéder à trois types de test. Le premier test constituera une comparaison de l'approche CLASS avec la combinaison de méthodes différentes : SPM (Significant Pattern Matching), N-Gram (avec N=3), SNN et KNN. Le deuxième test fera la comparaison de CLASS avec les algorithmes bayésiens naïfs et on utilisera pour ces deux premiers tests l'ensemble de données *"Enron"*. Le troisième test nous permettra d'évaluer les effets de l'apprentissage sur notre approche, on utilisera à cet effet l'ensemble de données "Ling-Spam".

Comparatif de CLASS avec différentes méthodes

Dans le tableau 3.1, nous récapitulons les résultats obtenus par CLASS en utilisant différentes méthodes SPM, N-Gram, SNN et KNN.

Pour évaluer les résultats, nous allons utiliser la Courbe ROC (Receiver Operating Charachteristic) et plus précisément la proportion des éléments en dessous de la courbe comme indice de qualité. Plus l'indice sera proche de 1, meilleure sera la classi-

fication. Le tableau 3.1 regroupe les résultats obtenus par l'expérimentation. Chaque colonne représente une combinaison de méthodes pour réaliser CLASS et chaque ligne représente les tests réalisés sur chacune des partitions de l'ensemble "Enron". La dernière ligne représente la moyenne des résultats obtenus pour chaque combinaison sur l'ensemble des partitions de Enron. Cette comparaison nous montre clairement que la combinaison SPM et SNN donne les meilleurs résultats avec une moyenne de 0.97. De ce fait, dans la suite de nos expériences, nous allons utiliser cette combinaison pour CLASS afin de réaliser la comparaison avec d'autres méthodes.

Comparatif de CLASS avec différents filtres bayésiens naïfs

Dans les expérimentations suivantes, nous allons comparer notre approche CLASS avec des filtres bayésiens naïfs :

- un filtre bayésien naïf Bernoulli multivarié introduit par *Sahami et al.* [38] ; comme on l'a vu dans le chapitre 2 avec des attributs booléens.
- un filtre bayésien naïf multinomial introduit par *Pantel et al.* [35] ; comme on l'a vu dans le chapitre 2 qui considère la fréquence d'apparition des termes.
- un filtre bayésien naïf multinomial introduit par *Schneider* [40] ; dans lequel les fréquences d'apparition des termes sont remplacées par des attributs booléens.
- un filtre bayésien naïf de Gauss multivarié introduit par *Rennie et al.* [37] ; qui est une variante d'un Bernoulli multivarié et qui utilise des attributs continus.
- un filtre bayésien flexible introduit par *John et al.* [14] ; dans lequel la distribution de chaque attribut est considérée comme la moyenne de plusieurs distributions normales.

Toutes ces approches sont détaillées et comparées par *Metsis et al.* [29].

Les tableaux 3.2 et 3.3 présentent dans chacune des colonnes le résultat du rappel [2] respectif des spams et courriels légitimes sur l'ensemble "Enron". Les résultats obtenus par notre approche sont très satisfaisants comparativement aux filtres bayésiens. Notre algorithme capte aussi bien le courrier légitime que le spam, le seul algorithme

2. $Rappel_i = \frac{documents\ correctements\ attribués\ à\ la\ classe\ i}{nombre\ de\ documents\ appartenant\ à\ la\ classe\ i}$

Algorithme	CLASS	Flexible	Gauss multivarié	Multinomial Fréquences des termes	Bernoulli Multivarié	Multinomial Attributs Booleens
Enron 1	95.77	90.50	93.08	95.66	97.08	96.00
Enron 2	96.13	93.63	95.80	96.81	91.05	96.68
Enron 3	96.76	96.94	97.55	95.04	97.42	96.94
Enron 4	97.45	95.78	80.14	97.79	97.70	97.79
Enron 5	99.32	99.56	95.42	99.42	97.95	99.69
Enron 6	98.05	99.55	91.95	98.08	97.92	98.10
Moyenne	97.25	95.99	92.32	97.13	96.52	97.53

tableau 3.2 – Pourcentage du rappel des spams sur le corpus 'Enron'

Algorithme	CLASS	Flexible	Gauss multivarié	Multinomial Fréquences des termes	Bernoulli Multivarié	Multinomial Attributs Booleens
Enron 1	96.67	97.64	94.83	94.00	93.19	95.25
Enron 2	97.61	98.83	96.97	96.78	97.22	97.83
Enron 3	97.02	95.36	88.81	98.83	75.41	98.88
Enron 4	99.67	96.61	99.39	98.30	95.86	99.05
Enron 5	97.32	90.76	97.28	95.65	90.08	95.65
Enron 6	95.19	89.97	95.87	98.08	82.52	96.88
Moyenne	97.25	94.86	95.53	95.12	89.05	97.26

tableau 3.3 – Pourcentage du rappel des courriels légitimes sur le corpus 'Enron'

qui performe un peu mieux que le nôtre est le bayésien naïf multinomial avec des attributs booléens.

Effet de l'apprentissage sur CLASS

Nous allons à travers le test suivant évaluer l'effet de l'apprentissage sur notre algorithme. Pour ce faire, nous utilisons l'ensemble de données "Ling-Spam" qui est partitionné en 10. À chaque nouveau test, nous rajoutons une partition supplémentaire dans l'étape d'apprentissage et nous classifions toujours la dernière partition. Les résultats en dessous de 50% de données d'apprentissage (5 partitions) sont faibles et peu représentatifs. Pour ce test, nous allons utiliser une nouvelle fois la Courbe ROC (Receiver Operating Characteristic) et la proportion des éléments en dessous de la courbe comme indice de qualité.

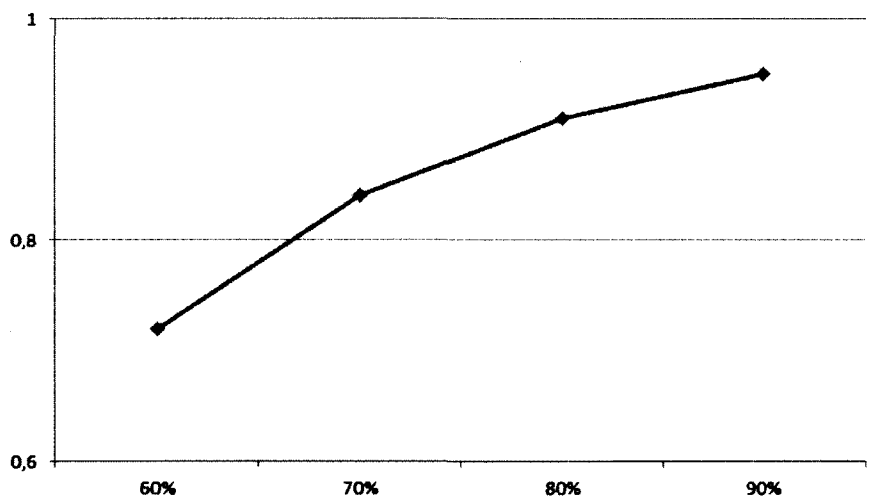

figure 3.5 – Classification avec CLASS en fonction de l'apprentissage

Le tableau 3.4 et le graphe de la figure 3.5 présentent les mêmes résultats. On voit sur ces résultats que notre approche est tributaire de l'apprentissage. Cependant, on constate que notre approche donne des résultats convenables dès 70% de taux d'apprentissage. De ce fait, on peut considérer que notre algorithme résiste bien au

Apprentissage	60%	70%	80%	90%
Indice de qualité	0.72	0.84	0.91	0.95

tableau 3.4 – Classification avec CLASS en fonction de l'apprentissage

changement de mots dans les spams. En effet, ce résultat est logique puisque nous ne captons pas des mots mais bien des motifs. À titre d'exemple, on considère les deux groupes de mots suivants : 'FREE VIAGRA' et 'FREE V1AGRA'. Les algorithmes se basant sur les mots vont découvrir seulement 1 mot commun entre ses deux groupes alors que notre approche trouvera les motifs communs suivants : 'FREE', 'REEV', 'AGRA'.

Synthèse

On peut conclure que notre approche en utilisant CLASS (SPM et SNN) donne de meilleurs résultats que plusieurs filtres bayésiens connus et qu'il réagit bien au morphisme des spams. Cependant, notre algorithme prend un temps considérable à son exécution du fait de la recherche des motifs significatifs dans de grands corpus. Dans le cas d'un déploiement sur un ordinateur personnel, on pourra proposer une solution qui va recompiler les motifs significatifs de manière programmée.

Conclusion

Dans ce mémoire, nous avons étudié les différentes approches et méthodes de détection de spam. Nous avons classifié ces méthodes en deux grandes catégories pour en étudier les différents aspects ; une première partie de cette étude s'est concentrée donc sur les différentes techniques et protocoles mis en place pour détecter les courriers indésirables. Ces dernières bien que nombreuses sont inefficaces individuellement et sont difficiles à mettre en œuvre aussi bien du côté des serveurs de courriel que du côté des utilisateurs de service de courriel. De plus, ces techniques sont pour la plupart délaissées par les webmails (hotmail, yahoo, etc.) qui sont une solution très utilisée par les utilisateurs de service de messagerie. Pour être efficace, ces techniques sont combinées entre elles et avec un filtre. Les filtres représentent la deuxième catégorie d'approches utilisées, ils sont en majorité basés sur une statistique bayésienne. Cette approche à l'avantage d'être évolutive, personnalisée et autonome (i.e. indépendante des serveurs et personnalisée à chaque utilisateur). Elle nécessite cependant une phase d'apprentissage qui peut être plus ou moins longue et qui va affecter grandement les résultats de la classification. Cette technique dépend également de la sélection des caractéristiques (tokens) ; elle va être dépendante du choix des séparateurs utilisés, de la sélection des tokens représentatifs et enfin du modèle de filtre bayésien utilisé. Ces différents choix et paramètres sont cruciaux pour obtenir de bons résultats.

Face à ce constat nous avons utilisé une nouvelle approche pour réaliser notre filtre. Cette approche, CLASS, a donné des résultats concluants dans la classification de différents types de données. Ces résultats sont disponibles dans "CLASS : A General Approach to Classifying Categorical Sequences" [16]. Elle se base sur la détection de motifs significatifs et sur une méthode de classification SNN. Cette méthode traite

les courriels non plus comme une suite de mots mais une séquence de caractères. Ainsi, avec cette méthode, nous n'avons plus à choisir les séparateurs et faire une sélection de caractéristiques. Ces deux étapes se réalisent au moment de la recherche des motifs significatifs. La classification quand à elle se fait grâce à SNN.

Les résultats des différentes expérimentations réalisées nous ont montré que notre approche donnait l'un des meilleurs résultats dans la classification. Ces résultats se traduisent par un grand pourcentage de rappel de courriers indésirables et légitimes. Dans notre dernière expérimentation, nous avons étudié la manière dont notre approche se comporte par rapport à l'apprentissage. Nous constatons ainsi que notre approche améliore également l'étape d'apprentissage et donc que notre approche s'adapte bien aux modifications de certains caractères pour détecter les motifs significatifs. Finalement, notre approche peut se combiner aux techniques de détection de spam qui sont présentées dans le chapitre 1 comme une alternative aux filtres bayésiens.

Annexe A

Le modèle OSI (Open Systems Interconnection)

Le modèle OSI (acronyme de "Open Systems Interconnection", "Interconnexion de systèmes ouverts") est un modèle de communication entre ordinateurs proposé par l'ISO (International Organization for Standardization). Il décrit les fonctionnalités nécessaires à la communication et l'organisation de ces fonctions. Il a pour but de caractériser et de normaliser les fonctions d'un système de communication en termes de couches d'abstraction. Les fonctions de communication similaires sont regroupés en couches logiques. Les documents décrivant le modèle OSI sont disponibles à partir de ITU-T comme les recommandations X.200-series (http ://www.itu.int/rec/T-REC-X.200-199407-I/fr). Ces spécifications présentent un modèle en 7 couches distinctes. Chaque couche bénéficiant des services de la couche directement inférieure et servant la couche supérieure. Ce modèle est illustré dans la figure A.1 et la fonction de chaque couche est spécifiée dans le tableau A.1.

Couche	Nom	Rôle
7	Applicative	C'est à ce niveau que sont les logiciels : navigateur, logiciel d'email, FTP, chat, *etc.*
6	Présentation	Elle est en charge de la représentation des données (de telle sorte qu'elle soit indépendante du type de microprocesseur ou du système d'exploitation par exemple) et - éventuellement - du chiffrement.
7	Session	En charge d'établir et maintenir des sessions (c'est à dire débuter le dialogue entre 2 machines : vérifier que l'autre machine est prête à communiquer, s'identifier, etc.)
4	Transport	En charge de la liaison d'un bout à l'autre. S'occupe de la fragmentation des données en petits paquets et vérifie éventuellement qu'elles ont été transmises correctement.
3	Réseau	En charge du transport, de l'adressage et du routage des paquets.
2	Liaison de données	En charge d'encoder (ou moduler) les données pour qu'elles soient transportables par la couche physique, et fournit également la détection d'erreur de transmission et la synchronisation.
1	Physique	C'est le support de transmissions lui-même : un fil de cuivre, une fibre optique, les ondes hertziennes, *etc.*

tableau A.1 – Description des couches du modèle OSI

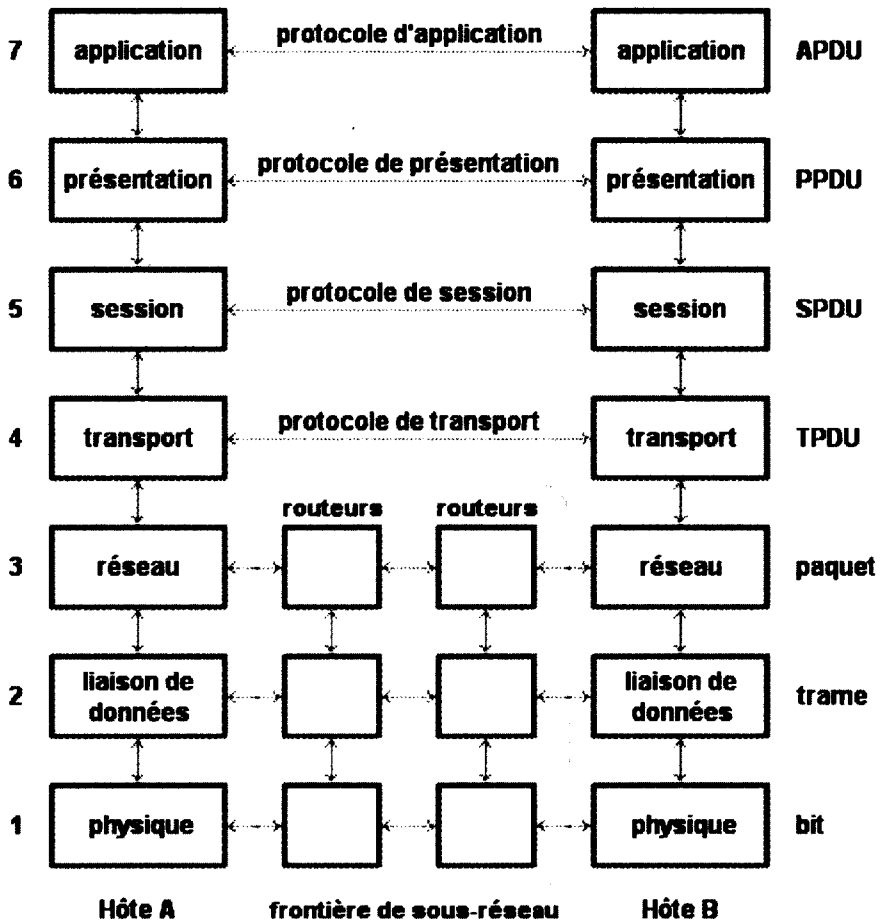

figure A.1 – Les 7 couches du modèle OSI

Annexe B

CRM114

CRM114 est un système pour examiner le courrier électronique entrant, le flux des journaux système, les fichiers de données ou d'autres flux de données, et trier, filtrer, ou modifier les fichiers entrants ou les flux de données en fonction des désirs de l'utilisateur. Les critères de classement des données peuvent se faire avec plusieurs méthodes, y compris les expressions régulières, les expressions rationnelles approchées, un modèle de Markov caché, Bayesian Chain Rule Orthogonal Sparse Bigrams, la corrélation, KNN / Hyperspace, Bit Entropy, CLUMP, SVM, les réseaux de neurones.

Les spams sont la cible principale de CRM114 mais cet outil n'est pas seulement utilisé dans ce domaine. Quand la plupart des algorithmes réalisent un filtre bayésien basé sur la fréquences des mots individuels dans les courriels, CRM114 obtient de meilleurs résultats dans la détection des spams en utilisant des séquences de mots pouvant aller jusqu'à une longueur de 5 mots. CRM114 a été utilisé pour trier des pages web, blog, fichiers logs, et bien d'autres type de données. La précision des résultats peut atteindre jusqu'à 99.9%. Tous les utilisateurs n'obtiennent pas de bons résultats avec le classificateur de base de CRM114, c'est pour cela que CRM114 a plusieurs type de classificateur. Il est facile de changer de classificateur et d'exécuter un script permettant de voir ce que ce changement apporte en terme de vitesse d'exécution, precision, espace disque utilisé, taux d'apprentissage, *etc.*. CRM114 est développé sous licence GPL.

Bibliographie

[1] E. ALLMAN, J. CALLAS, M. DELANY, M. LIBBEY, J. FENTON et M. THOMAS. « DomainKeys Identified Mail (DKIM) Signatures ». URL http ://www. ietf. org/rfc/rfc4871. txt accessed 2012-05-21, 2007.

[2] M.W. BERRY et R.D. FIERRO. « Low-Rank Orthogonal Decompositions for Information Retrieval Applications ». Numerical Linear Aalgebra with Applications, 3(4) :301–327, 1996.

[3] Olivier BOUSQUET. « Introduction to Statistical Learning Theory ». Biological Cybernetics, 3176(1) :169–207, 2004.

[4] William B CAVNAR et John M TRENKLE. « N-Gram-Based Text Categorization ». Ann Arbor MI, 48113(2) :161–175, 1994.

[5] C.Y. CHIU et Y.T. HUANG. « Integration of Support Vector Machine with Naïve Bayesian Classifier for Spam Classification ». Industrial Engineering, 1 :618–622, 2007.

[6] Lorrie Faith CRANOR et Brian A. LAMACCHIA. « Spam! ». Commun. ACM, 41(8) :74–83, août 1998.

[7] Scott DEERWESTER, Susan T DUMAIS, George W FURNAS, Thomas K LANDAUER et Richard HARSHMAN. « Indexing by Latent Semantic Analysis ». Journal of the American Society for Information Science, 41(6) :391–407, 1990.

[8] Peter Edward DUDA, Richard O. Hart. « Pattern Classification and Scene Analysis », volume 2, pages 10–43. John Wiley & Sons Inc, 1 édition, 1973.

[9] William A GALE et Kenneth W CHURCH. « What is Wrong with Adding One ? », pages 189–200. Rodopi, 1 édition, 1994.

[10] I.J. GOOD. « The Population Frequencies of Species and the Estimation of Population Parameters ». Biometrika, 40(3-4) :237–264, 1953.

[11] P. GRAHAM. « A Plan for Spam ». URL http ://www. paulgraham. com/spam. html accessed 2012-05-21, 2002.

[12] José M GÓMEZ HIDALGO, Manuel Maña LÓPEZ et Enrique Puertas SANZ. « Combining Text and Heuristics for Cost-Sensitive Spam Filtering ». Numéro 99, page 99–102. Association for Computational Linguistics, 2000.

[13] P. HOFFMAN. « SMTP Service Extension for Secure SMTP over Transport Layer Security ». URL http ://www. ietf. org/rfc/rfc3207. txt accessed 2012-05-21, 2002.

[14] G H JOHN et P LANGLEY. « Estimating Continuous Distributions in Bayesian Classifiers ». Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, 1 :338–345, 1995.

[15] S KARLIN et G GHANDOUR. « Comparative Statistics for DNA and Protein Sequences : Single Sequence Analysis ». Proceedings of the National Academy of Sciences of the United States of America, 82(17) :5800–5804, 1985.

[16] A. KELIL, A. NORDELL-MARKOVITS, P.O.Y. ZARALAHY et S. WANG. « CLASS : A General Approach to Classifying Categorical Sequences ». Canadian Journal of Electrical and Computer Engineering, 34(4) :158–166, 2009.

[17] Abdellali KELIL et Shengrui WANG. « SCS : A New Similarity Measure for Categorical Sequences », pages 498–505. 2008.

[18] B. KLIMT et Y. YANG. « The Enron Corpus : A New Dataset for Email Classification Research ». pages 21–26, Berlin, Germany, 2004.

[19] R. KOHAVI et G.H. JOHN. « Wrappers for Feature Subset Selection ». Artificial Intelligence, 97(1-2) :273–324, 1997.

[20] Grzegorz KONDRAK. « N-gram Similarity and Distance ». volume 3772, pages 115–126. Springer, 2005.

[21] M. KUCHERAWY. « Authentication-Results Registration Update for Sender Policy Framework (SPF) Results ». URL http ://www. ietf. org/rfc/rfc6577. txt accessed 2012-05-21, 2012.

[22] N. KUSHMERICK. « Wrapper Induction for Information Extraction ». Thèse de doctorat, University of Washington, 1997.

[23] N. KUSHMERICK. « Wrapper Induction : Efficiency and Expressiveness ». Artificial Intelligence, 118(1-2) :15–68, 2000.

[24] T.K. LANDAUER, P.W. FOLTZ et D. LAHAM. « An Introduction to Latent Semantic Analysis ». Discourse Processes, 25(2) :259–284, 1998.

[25] V.I. LEVENSHTEIN. « Binary Codes Capable of Correcting Deletions, Insertions, and Reversals ». Dans Soviet Physics Doklady, volume 10, pages 707–710, 1966.

[26] J. LEVINE. « DNS Blacklists and Whitelists ». URL http ://www. ietf. org/rfc/rfc5782. txt accessed 2012-05-21, 2010.

[27] John R LEVINE. « Experiences with Greylisting », pages 4–5. Citeseer, 2005.

[28] Andrew MCCALLUM et Kamal NIGAM. « A Comparison of Event Models for Naive Bayes Text Classification ». Dimension Contemporary German Arts And Letters, 752 :41–48, 1998.

[29] Vangelis METSIS, Ion ANDROUTSOPOULOS et Georgios PALIOURAS. « Spam Filtering with Naive Bayes - Which Naive Bayes ». Third Conference on Email and Antispam CEAS, 17 :125–134, 2006.

[30] F. MHAMDI, R. RAKOTOMALALA et M. ELLOUMI. « A Hierarchical n-Grams Extraction Approach for Classification Problem ». Advanced Internet Based Systems and Applications, 4879 :211–222, 2009.

[31] HB MITCHELL et PA SCHAEFER. « A 'Soft' K-Nearest Neighbor Voting Sscheme ». International Journal of Intelligent Systems, 16(4) :459–468, 2001.

[32] Dunja MLADENIC et Marko GROBELNIK. « Feature Selection for Unbalanced Class Distribution and Naive Bayes », pages 258–267. Morgan Kaufmann, San Francisco, CA, 1999.

[33] S.B. NEEDLEMAN et C.D. WUNSCH. « A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of two Proteins ». Journal of Molecular Biology, 48(3) :443–453, 1970.

[34] H. NEY, U. ESSEN et R. KNESER. « On Structuring Probabilistic Dependences in Stochastic Language Modelling ». Computer Speech and Language, 8(1) :1–38, 1994.

[35] P PANTEL et D LIN. « SpamCop : A Spam Classification and Organization Program », pages 1–8. 1998.

[36] B. RAMSDELL et S. TURNER. « Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification ». URL http ://www. ietf. org/rfc/rfc5751. txt accessed 2012-05-21, 2010.

[37] Jason D M RENNIE, Lawrence SHIH, Jaime TEEVAN et David R KARGER. « Tackling the Poor Assumptions of Naive Bayes Text Classifiers ». Machine Learning, 20(1973) :616–623, 2003.

[38] M. SAHAMI, S. DUMAIS, D. HECKERMAN et E. HORVITZ. « A Bayesian Approach to Filtering Junk E-Mail ». Dans Learning for Text Categorization : Papers from the 1998 Workshop, volume 62, pages 98–05. Madison, Wisconsin : AAAI Technical Report WS-98-05, 1998.

[39] G. SAKKIS, I. ANDROUTSOPOULOS, G. PALIOURAS, V. KARKALETSIS, C.D. SPYROPOULOS et P. STAMATOPOULOS. « A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists ». Information Retrieval, 6(1) :49–73, 2003.

[40] Karl-Michael SCHNEIDER. « On Word Frequency Information and Negative Evidence in Naive Bayes Text Classification ». Advances in Natural Language Processing, Volume 323 :474–485, 2004.

[41] R. SIEMBORSKI et A. MELNIKOV. « SMTP Service Extension for Authentication ». URL http ://www. ietf. org/rfc/rfc4954. txt accessed 2012-05-21, 2007.

[42] W. SONG et S.C. PARK. « Latent Semantic Analysis for Vector Space Expansion and Fuzzy Logic-Based genetic clustering ». Knowledge and Information Systems, 22(3) :347–369, 2010.

[43] V.N. VAPNIK. The Nature of Statistical Learning Theory. Springer Verlag, 2 édition, 1999.

[44] V.N. VAPNIK. « An Overview of Statistical Learning Theory ». Neural Networks, IEEE Transactions on, 10(5) :988–999, 1999.

[45] I.H. WITTEN et T.C. BELL. « The Zero-Frequency Problem : Estimating the Probabilities of Novel Events in Adaptive Text Compression ». Information Theory, IEEE Transactions on, 37(4) :1085–1094, 1991.

[46] Yiming YANG et Jan O PEDERSEN. « A Comparative Study on Feature Selection in Text Categorization ». Methods, 20(15) :412–420, 1997.

[47] W.S. YERAZUNIS. « The Spam-Filtering Accuracy Plateau at 99.9% Accuracy and How to Get Past It ». URL http ://www. merl. com/reports/docs/TR2004-091. pdf accessed 2012-05-21, 2004.

[48] J.A. ZDZIARSKI. Ending Spam : Bayesian Content Filtering and the Art of Statistical Language Classification. No Starch Press, 1 édition, 2005.