CLASSIFICATION THÉMATIQUE DE COURRIELS AVEC APPRENTISSAGE SUPERVISÉ, SEMI SUPERVISÉ ET NON SUPERVISÉ

Rémy KESSLER, Juan Manuel TORRES-MORENO et Marc EL-BEZE

kessler@univ-avignon.fr, torres@univ-avignon.fr, elbeze@univ-avignon.fr

Laboratoire d'Informatique d'Avignon - Université d'Avignon et des Pays de Vaucluse
339 chemin des Meinajaries, Agroparc, BP 1228, 84911 Avignon Cedex 9, France
Tel. : +33 (0) 4 90 84 35 09

Résumé : Les nouvelles formes de communication écrite (courriers électroniques, forums, *chats*, SMS, etc.) présentent des défis considérables pour leur traitement automatique. Nous présentons des recherches destinées à créer des outils et des ressources génériques pour la classification de courriels. La capacité d'une entreprise de gérer efficacement, rapidement et à moindre coût, ces flux d'informations devient un enjeu majeur pour la satisfaction des clients. Ceci nécessite, en particulier, de disposer d'outils informatiques permettant notamment le routage pour acheminer les courriels vers le destinataire concerné et l'automatisation de réponses. Nous nous attachons à traiter dans cette étude des problèmes posés par le routage précis de courriels. Après un processus puissant de filtrage et de lemmatisation, nous utilisons la représentation vectorielle de textes avant d'effectuer la classification par des approches supervisées, semi supervisées et non supervisées. Nous avons trouvé, par ailleurs, une initialisation semi supervisée qui optimise l'apprentissage non supervisé. Lors des tests préli-minaires, nous avons obtenu de très bonnes performances sur des corpus réalistes.

Mots-clés : apprentissage supervisé et non-supervisé, machines à vecteurs support (*SVM*), *fuzzy k-means*, classification de textes, routage automatique de courriels.

Keywords: supervised and unsupervised learning, Support Vector Machines, *fuzzy k-means*, Text classification, automatic e-mail routing.

Palabras clave: aprendizaje supervisado y no supervisado, máquinas de soporte vectorial (*SVM*), *fuzzy k-means*, clasificación de textos, rutaje automático de correo electrónico.

1 Introduction

Les nouvelles formes de communication écrite posent des défis considérables aux systèmes de traitement automatique de la langue car on observe des phénomènes linguistiques bien particuliers comme les émoticônes[1], les acronymes, les fautes (orthographiques, typogra-phiques, mots collés, etc.) d'une très grande morpho-variabilité et d'une créativité explosive. Ces phénomènes doivent leur origine au mode de communication (direct ou semi direct), à la rapidité de composition du message ou aux contraintes technologiques de saisies imposées par le matériel (terminal mobile, téléphone, etc.). Dans cet article, nous désignons par **phonécriture** ou **phonécrit** toute forme écrite qui utilise un type d'écriture phonétique sans contraintes ou avec des règles établies par l'usage[2]. Le traitement automatique des courriels est extrêmement difficile à cause de son caractère imprévisible [1,9] : des textes trop courts (mo-yenne de 11 mots par courriel), régis par une syntaxe pauvre ou mal orthographiés. Ceci impose donc d'utiliser des outils de traitement automatique robustes et flexibles. Les méthodes d'apprentissage automatique à partir de textes (fouille de documents), permettent d'apporter des solutions partielles aux tâches évoquées. Elles semblent bien adaptées aux applications de filtrage, de routage, de recherche d'information, de classification thématique et de structuration non supervisée de corpus. Ces méthodes présentent de surcroît l'intérêt de fournir des réponses adaptées à des situations où les corpus sont en constante évolution ou bien contiennent de l'information dans des langues étrangères. L'objectif de cette recherche consiste à proposer l'application des méthodes d'apprentissage afin d'effectuer la classification automatique de courriels visant leur routage, combinant techniques probabilistes et *Support Vector Machines* (SVM). La catégorisation thématique est au coeur de nombreuses applications de traitement de la langue. Ce contexte fait émerger un certain nombre de questions théoriques nouvelles, en particulier en relation avec la problématique du traitement d'informations textuelles incom-plètes et/ou très bruitées.

2 Position du problème

On se place dans le cas où une boîte aux lettres reçoit un grand nombre de courriels correspondant à plusieurs thématiques. Une personne doit lire ces courriers et les rediriger vers le service concerné (les courriels de problèmes techniques vers le service technique, ceux pour le service après vente seront redirigés vers ce dernier, etc.). Il s'agit donc de développer un système pour automatiser cette tâche.

Après une recherche sur Internet, il s'est avéré difficile de trouver des corpus de courriels en français (des corpus anglais existent cependant pour de la classification de *spams*). Nous avons donc décidé de créer une adresse électronique et de l'abonner à diverses listes de diffusion[3] où *newsletters*[4] de thèmes variés. Les corpus réalistes qui ont été ainsi collectés présentent un certain nombre de caractéristiques particulières qui sont reportées dans le tableau 1.

Nb. total de courriels	P=1000		Nb. d'auteurs différents	247	
Nb. total de mots bruts	10016		Auteurs ayant émis moins de 2 courriels	69	28%
Nb. de courriels avec pièce jointe	150	15%	Auteurs ayant émis entre 2 et 5 courriels	125	51%
Nb. de courriels court (< 11 mots)	665	67%	Auteurs ayant émis entre 5 et 10 courriels	23	9%
Nb. de courriels long (> 11 mots)	335	33%	Auteurs ayant émis plus de 10 courriels	30	12%
Taille moyenne d'un courriel en mots				11	

Tableau 1 Statistiques du corpus

3 Méthodes

Les méthodes que nous avons retenues reposent sur la représentation vectorielle de textes, qui, même si elle est très différente d'une analyse structurale linguistique, s'avère performante et rapide [10]. Ces

[1] Symboles utilisés dans les messages pour exprimer les émotions, exemple le sourire :-) ou la tristesse :-(
[2] Par exemple kdo à la place de cadeau, a+ pour à plus, 10ko pour dictionnaire, etc.
[3] Football, jeux de rôles, ornithologie, cinéma, jeux vidéo, poème, humour, etc.
[4] Sécurité informatique, journaux, matériel informatique, etc.

méthodes ont par ailleurs la propriété d'être assez indépendantes de la langue[5]. La première étape consiste à nettoyer le corpus afin de séparer l'en-tête, le corps et la pièce jointe du courrier électronique[6]. Bien que l'en-tête contienne des meta-informations qui pourraient être utilisées pour la tâche de classification. Ensuite, nous réalisons un prétraitement où des processus puissants de filtrage et de lemmatisation sont déclenchés afin de réduire la dimensionnalité des matrices [13, 14, 15]. La lemmatisation (même si elle est plus coûteuse en temps) et non un simple *stemming* s'avère plus performante pour le français, qui est une langue latine à fort taux de flexion [6]. Au cours de cette phase, en vue de réduire cette dimensionnalité, nous effectuons différents traitements que nous détaillerons par la suite. Nous obtenons donc à la fin de ce prétraitement une matrice qui sera divisée aléatoirement en sous-ensembles d'apprentissage et de généralisation, puis traitée par les méthodes d'apprentissage. Nous avons décidé d'utiliser les algorithmes *k*-means et fuzzy *k*-means pour l'apprentissage non supervisé et SVM pour l'apprentissage supervisé. La figure 1 illustre l'ensemble des opérations.

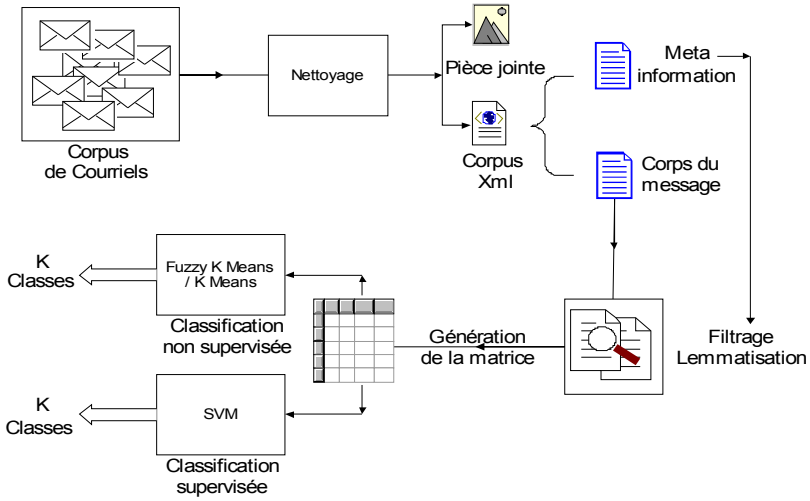


Figure 1 Chaîne de traitement de classification hybride de courriels

3.1 Prétraitement

La première partie du prétraitement a consisté à identifier dans le corpus chaque courriel différent, séparer le corps du message des pièces jointes. Cette première étape génère un fichier présenté, au format XML afin d'en faciliter le parcours. Une fois cette tâche réalisée, un second processus se charge d'effectuer les traitements de filtrage. Ainsi on supprime la micro publicité (*microspams*) qui n'apporte aucune information permettant de catégoriser le courriel mais, au contraire, ajoute du bruit risquant de gêner cette catégorisation. Il s'agit en général de publicités ajoutées au bas des courriels par les fournisseurs de service de messagerie électronique comme le montre l'exemple suivant :

___[Pub]_____
Inscrivez-vous gratuitement sur Tandaime, Le site de rencontres !
http://rencontre.rencontres.com/index.php?origine=4

Nous nous sommes attachés à supprimer la publicité générique des courriels, celle-ci étant généralement précédée d'une ligne composée de la façon suivante `____[ Pub ]_____`, ou encore `**********`. La particularité de ces lignes a permis de les enlever sans risque de perte d'informations au niveau du corps du message. Nous avons par la suite supprimé la micro publicité propre au corpus, celle-ci se présentant, la plupart du temps, sous la forme de liens HTML vers des pages Internet Yahoo. À l'aide d'un dictionnaire constitué à partir de sites[7] et décrivant les divers termes

[5] Pour l'instant, nos tests sont limités au français.
[6] Notre classification s'effectue pour l'instant uniquement sur le corps du message.
[7] http://www.mobimelpro.com/portail/fr/my/dictionnaire_sms.asp
http://www.mobilou.org/10kosms.htm
http://www.affection.org/chat/dico.html

de phonécriture, nous remplaçons ceux-ci par leurs équivalents en langue française. Cette étape de "traduction" est réalisée avant la suppression de la ponctuation car beaucoup de termes phonécrits sont composés à l'aide de ponctuation (:) → sourire, A+ →à plus tard, @2m1 → à demain). Ensuite nous appliquons les processus classiques de traitement de la langue :

Filtrage : Nous effectuons dans un premier temps une suppression des verbes et des mots fonctionnels (*être, avoir, pouvoir, falloir ...*), des expressions courantes (*par exemple, c'est-à-dire, chacun de ...*), de chiffres (numériques et/ou textuelles) et des symboles comme $, #, *, etc. venant bruité la classification. Par la suite nous identifions les mots composés qui représentent souvent un concept bien spécifique. Tous ces mots peuvent être répétés ou non. Il est important de définir si on travaille sur des formes fléchies ou des formes de base[8]. Chacun de ses traitements permet de réduire la complexité de la matrice.

Lemmatisation : Ce traitement entraîne une réduction importante du lexique. La lemmatisation simple consiste à trouver la racine des verbes fléchis et à ramener les mots pluriels et/ou féminins au masculin singulier[9] avant de leur associer un nombre d'occurrences. Ce processus permet d'amoindrir la malédiction dimensionnelle[10] qui pose de très sérieux problèmes de représentation dans le cas des grandes dimensions. La lemmatisation permet donc de diminuer le nombre de termes qui définiront les dimensions de l'espace vectoriel. D'autres mécanismes de réduction du lexique sont aussi déclenchés. Ainsi, les mots composés sont repérés à l'aide d'un dictionnaire, puis transformés en un terme unique lemmatisé[11].

Représentation vectoriel : Dans une tâche de classification, chaque vecteur Γ^μ appartient à une classe $\tau^\mu = g(\Gamma^\mu)$, étant g une fonction inconnue. L'ensemble d'exemples dont on dispose, appelé l'ensemble d'apprentissage, consiste en P couples d'entrée-sortie $(\overrightarrow{\Gamma^\mu}, \tau^\mu); \mu=1,..,P$ où les sorties τ^μ (les classes) sont connues. Nous dénoterons cet ensemble $\Lambda = g\{(\overrightarrow{\Gamma^\mu}, \tau^\mu); \mu=1,..,P\}$. Si l'on fait un apprentissage supervisé, un classifieur a besoin de connaître les classes τ^μ afin d'approcher la fonction g. Dans l'apprentissage non supervisé, τ^μ peut être ignoré, car on fait un regroupement des objets en fonction uniquement de leurs caractéristiques $\overrightarrow{\Gamma^\mu}$. Avec le prétraitement, nous transformons le corpus en un ensemble de P vecteurs (nombre de courriel) à N dimensions (taille du lexique). Nous obtenons donc une matrice fréquentielle $\Gamma = (\overrightarrow{\Gamma^\mu}); \mu = 1,..., P$ où chaque composante $\overrightarrow{\Gamma^\mu} = (\overrightarrow{\Gamma_1^\mu}, \overrightarrow{\Gamma_2^\mu}, ..., \overrightarrow{\Gamma_N^\mu})$ contient la fréquence $\overrightarrow{\Gamma_i^\mu}$ du terme i dans un courriel μ.

$$\mathbf{\Gamma} = \begin{bmatrix} \Gamma_1^1 & \Gamma_2^1 & \Gamma_3^1 & \dots & \Gamma_i^1 & \dots & \Gamma_N^1 \\ \Gamma_1^2 & \Gamma_2^2 & \Gamma_3^2 & \dots & \Gamma_i^2 & \dots & \Gamma_N^2 \\ \Gamma_1^3 & \Gamma_2^3 & \Gamma_3^3 & \dots & \Gamma_i^3 & \dots & \Gamma_N^3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \Gamma_1^\mu & \Gamma_2^\mu & \Gamma_3^\mu & \dots & \Gamma_i^\mu & \dots & \Gamma_N^\mu \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \Gamma_1^P & \Gamma_2^P & \Gamma_3^P & \dots & \Gamma_i^P & \dots & \Gamma_N^P \end{bmatrix}, \quad \Gamma_i^\mu \in \{0, 1, 2, \dots\} \qquad (1)$$

3.2 Observation de la matrice

[8] C'est pourquoi on emploie plutôt la notion de *terme* pour désigner un mot plus abstrait.
[9] Ainsi on pourra ramener à la même forme **chanter** les mots *chante, chantaient, chanté, chanteront* et éventuellement *chanteur*
[10] *The curse of dimensionality*
[11] *pomme de terre* et *pommes de terre* deviennent ainsi **pomme_de_terre**

La figure 2 présente une répartition des termes en fonction des courriels. Les classes ont été mélangées de façon aléatoire lors de la création, ce qui explique la division de certaines classes. L'axe des ordonnées est la liste des termes pour le corpus tandis que l'axe des abscisses représente la liste des courriels. Sur ce graphique, on observe que la catégorie 4 est divisée en deux parties (courriels 1 à 25 puis 100 à 125). La densité des 1000 premiers termes redevient importante entre le 100ème et le 125ème termes, mais ailleurs elle reste assez faible. Ceci est observable aussi pour la catégorie 2 (entre le courriel 25 et 50 puis entre 125 et 150) ou l'on remarque très bien l'absence de termes des catégories 3 et 4 dans le second morceau (faible densité des termes 1500 à 2500). On observe par ailleurs que lors de l'apparition d'une nouvelle catégorie, les nouveaux termes sont en forte densité dès le début de celle-ci.

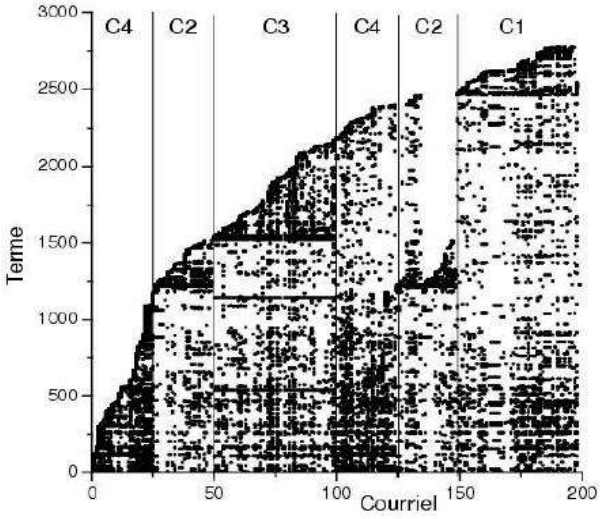


Figure 2 Représentation des termes en fonction des courriels

3.3 Mesure des performances en apprentissage et en généralisation

Soit Λ un ensemble de P exemples avec leur classe d'appartenance τ^{μ}. Soit γ_1 un sous-ensemble d'apprentissage de P_1 exemples tiré au hasard, et γ_2 le sous-ensemble de test indépendant de P_2 exemples, tel que $\Lambda = \gamma_1 \cup \gamma_2$ et $P = P_1 + P_2$. Un exemple mal classé par un classifieur est un exemple dont la classe attribuée τ^{μ} n'est pas correcte. Ce type d'erreur peut être mesuré en apprentissage ε_a ou en généralisation ε_g. Ce sont des valeurs comprises entre 0 et 1, ou exprimées en pourcentage. Pour mesurer la performance des algorithmes, nous avons utilisé les expressions suivantes afin d'estimer ε_a et ε_g :

$$\varepsilon_a = \frac{Nb.\ exemples \in \gamma_1\ mal\ classés}{card\{\gamma_1\}} \quad , \quad \varepsilon_g = \frac{Nb.\ exemples \in \gamma_2\ mal classés}{card\{\gamma_2\}} \quad ;$$

où *card*{•} représente le nombre d'éléments d'un ensemble. Cette performance sera utilisée dorénavant dans nos tests.

3.4 Apprentissage non supervisé avec fuzzy k-means et k-means

L'algorithme Fuzzy *k*-means [2, 5] permet d'obtenir un regroupement des éléments par une approche floue avec un certain degré d'appartenance, où chaque élément peut appartenir à une ou plusieurs classes, à la différence de *k*-means, où chaque exemple appartient à une seule classe (partition dure). Fuzzy *k*-means minimise la somme des erreurs quadratiques avec les conditions suivantes :

$$\sum_{k=1}^{c} m_{\mu k} = 1 \ , \ \mu = 1,2,\cdots,P \ ;$$

(2)

$$\sum_{\mu=1}^{P} m_{\mu k} > 0 \ , k = 1,2,\cdots, c$$

(3)

$$m_{\mu k} \in [0,1] \ \mu = 1,2,\cdots, P; \ k = 1,...,c$$

(4)

On définit la fonction objective : $J = \sum_{\mu=1}^{P}\sum_{k=1}^{c} m_{\mu k}^{f} d^{\lambda}(\Gamma^{\mu}, \beta^{k})$

(5)

où P est le nombre de données dont on dispose, c est le nombre de classes désiré, β^k est le vecteur qui représente le centroïde (barycentre) de la classe k, Γ^{μ} est le vecteur qui représente chaque exemple μ et $d^{\lambda}(\Gamma^{\mu}, \beta^{k})$ est la distance entre l'exemple Γ^{μ} et β^k en accord avec une définition de distance (voir 3.4.1) et que nous écrirons $d_{\mu k}^{\lambda}$ afin d'alléger la notation. f est le paramètre flou, valeur comprise dans l'intervalle [2, ∞) qui détermine le degré de *flou* de la solution finale, contrôlant le degré de recouvrement entre les classes. Avec f =1, la solution devient une partition dure. Si $f \to \infty$ la solution approche le maximum de *fuzzyfication* et toutes les classes risquent de se confondre en une seule. La minimisation de la fonction objective J fournit la solution pour la fonction d'appartenance $m_{\mu k}$ (4) :

$$m_{\mu k} = \frac{d_{\mu k}^{\lambda/(f-1)}}{\sum_{j=1}^{c} d_{\mu j}^{\lambda/(f-1)}} \quad \mu=1,2,...,P; \ k=1,...,c \ ;$$

(6)

$$\beta^{k} = \frac{\sum_{\mu=1}^{P} m_{\mu k}^{f} \Gamma^{\mu}}{\sum_{\mu=1}^{P} m_{\mu k}^{f}} \qquad k = 1,2,..., c \qquad (7)$$

L'algorithme *fuzzy k-means* est donc le suivant :

1. Fixer le nombre k de classes $1 < k < c$;
2. Fixer une valeur du paramètre flou $f > 2$;
3. Choisir une définition de distance adéquate $d_{\mu k}^{\lambda}$ (voir la sous-section 3.4.1) ;
4. Fixer un (petit) critère d'arrêt ε ;
5. Initialiser $m_{\mu k}$ (voir la sous-section 3.4.2) ;
6. Pour chaque itération recalculer β^k en utilisant l'équation (7) et les valeurs $m_{\mu k}$ de l'itération précédente ;
7. Recalculer $m_{\mu k}$ en utilisant l'équation (6) et les valeurs de β^k de l'itération précédente ;
8. Si la différence absolue de $m_{\mu k}$ entre deux itérations < ε alors stop sinon aller à 6.

L'intérêt d'utiliser *fuzzy k-means* dans le cadre de la classification thématique de courrier électronique consiste à router un message vers un destinataire prioritaire (celui avec le degré d'appartenance le plus élevé) et en copie conforme (Cc) ou caché (Bcc) vers celui (ou ceux) dont le degré d'appartenance

dépasse un certain seuil établi à l'avance.

3.4.1 Calcul de distance entre les vecteurs

Afin d'effectuer la classification, nous calculons la distance entre les vecteurs et les centroïdes. Nous avons utilisé pour cela la distance de Minkowski :

$$d^{\lambda}(a,b) = (\sum_{i=1}^{N} |a_i - b_i|^{\lambda})^{1/\lambda} \quad (8)$$

Nous avons fait une implémentation de l'algorithme avec λ=1 (distance de *Manhattan*), cependant les résultats obtenus étant décevants, et nous avons utilisé λ=2 distance euclidienne) avec $d^2(\Gamma^{\mu}, \beta^{k}) \leftarrow \left\| \Gamma^{\mu} - \beta_k \right\|$. La distance de *Manhattan* permet principalement de faire la différence entre la présence ou l'absence d'un terme *i* dans un courriel , et la distance euclidienne apporte en plus les poids de chaque terme, ce qui permet de mieux classer les exemples.

3.4.2 Initialisation aléatoire ou semi-supervisée ?

On sait que *k*-means et fuzzy *k*-means sont des algorithmes performants mais fortement dépendants de l'initialisation [14]. On est donc confronté au problème de l'initialisation des centroïdes β^k. Nous avons d'abord testé la méthode avec des initialisations aléatoires, mais l'erreur d'apprentissage, ε_a, était autour de 25% dans le meilleur des cas (voir figure 6). De même l'erreur en généralisation, ε_g était toujours assez importante. Ceci est dû au fait que l'algorithme semble piégé dans des minimums locaux. Nous avons donc décidé d'initialiser de façon semi supervisée en prenant un petit nuage d'exemples (avec leur classe) afin d'avoir des points de départ mieux situés pour nos centroïdes. Nous avons fait une étude de cette initialisation semi supervisée. Sur la figure 6, sont illustrés les résultats que nous avons obtenus sur 10 ensembles d'apprentissage tirés au hasard. L'initialisation semi supervisée a donc résolu le problème. Il est cependant important de rappeler que l'apprentissage avec *k*-means est toujours non supervisé, et qu'il suffit d'initialiser avec un nombre d'exemples entre 10 et 20% pour obtenir $\varepsilon_a < 10\%$.

Nous avons voulu par ailleurs, connaître l'incidence du paramètre de flou *f* afin d'améliorer les résultats. Nous avons donc effectué une série de tests en ne faisant varier que ce paramètre, *f* allant de 2 à 50. Les résultats de la figure 5 montrent qu'au delà d'une valeur de 10, les variations sur ε_a sont négligeables.

3.5 Machines à support vectoriel

Ces machines, proposées par Vapnik [15] ont été utilisées avec succès dans plusieurs tâches d'apprentissage et sont actuellement en plein essor. Elles offrent en particulier une bonne approximation du principe de minimisation du risque structurel. La méthode repose sur les idées suivantes :

- les données sont projetées dans un espace de grande dimension par une transformation basée sur un noyau linéaire, polynomial ou gaussien comme le montre la figure 3.
- dans cet espace transformé, les classes sont séparées par des classifieurs linéaires qui maximisent la marge (distance entre les classes).
- les hyperplans peuvent être déterminés au moyen d'un nombre de points limités qui seront appelés les « vecteurs supports ».

La complexité d'un classifieur SVM va donc dépendre non pas de la dimension de l'espace des données, mais du nombre de vecteurs supports nécessaires pour réaliser la séparation. Les SVM ont déjà été appliqués au domaine de la classification du texte dans plusieurs travaux [7, 16], mais toujours en utilisant des corpus bien rédigés (des articles journalistiques, scientifiques...). Nous avons choisi de les

utiliser dans ce type de corpus particulier.

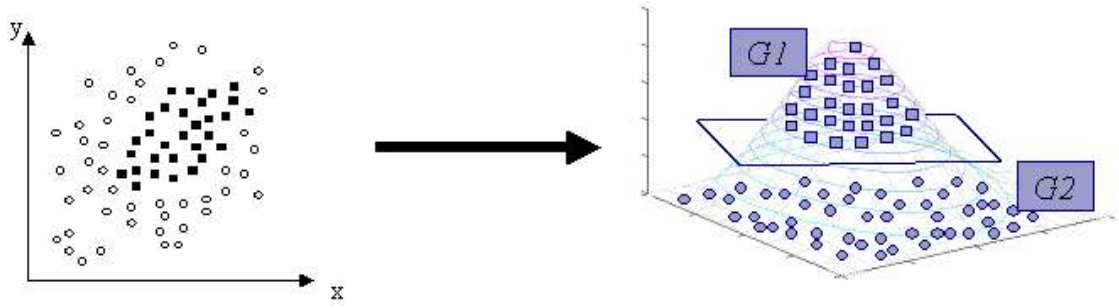


Figure 3 En plongeant les données dans un espace de plus grande dimension, on simplifie la tâche de classification

3.5.1 Implémentation

Nous avons testé plusieurs implémentations différentes des machines à support vectoriel (lia_sct[12], SVMTorch[13], Winsvm[14], M-SVM[15]) afin de pouvoir utiliser la plus efficace. Nous avons finalement décidé d'utiliser une implémentation de l'algorithme de Collobert [4], SVMTorch, qui permet une approche multi-classes des problèmes de classification. Celle-ci utilise le principe *One-against-the-Rest*, où chaque classe est comparée aux à l'ensemble des autres afin de trouver l'hyperplan séparateur. On utilise pour cela une fonction noyau qui permet de projeter les données dans un espace de grande dimension de la façon suivante : sous les conditions de Mercer [15], le produit scalaire dans le nouvel espace peut être réécrit au moyen d'une fonction noyau *kernel* $K(a,b)$ telle que $K(a,b) = (\Phi(a) \bullet (\Phi(b))$. SVMTorch permet de tester plusieurs types de fonctions noyaux :

- un noyau simple (polynôme de premier degré) ;
- un noyau polynomial de degré $d(a,b) \rightarrow (a,b)^d$;
- une gaussienne à base radiale (FBR) $(a,b) \rightarrow \exp\left(-\frac{\|a-b\|^2}{2\sigma^2}\right)$;
- un noyau basé sur une forme particulière de réseau de neurones avec fonctions d'activation sigmoïdales $(a,b) \rightarrow \tanh(sa \cdot b + r)$

Les résultats des tests que nous avons effectués ont été obtenus à l'aide d'une fonction à noyau simple. Nous prévoyons aussi de tester d'autres fonctions noyaux afin de savoir si cela influence positivement nos résultats.

3.6 La méthode hybride

Nous avons décidé de combiner les deux méthodes d'apprentissage afin d'avoir les avantages de chacune d'entre elles. En effet, l'apprentissage non supervisé avec *k*-means donnait de bons résultats lors de la phase d'apprentissage mais faisait beaucoup d'erreurs en généralisation. D'un autre côté,

[12] lia_sct:*a decision-tree based string classifier from F. Béchet*
[13] *http ://www.idiap.ch/*
[14] *http://liama.ia.ac.cn/PersonalPage/lbchen/winsvm.htm*
[15] *http://www.loria.fr/~guermeur/*

l'apprentissage avec les SVM est supervisé donc coûteux. Ainsi, nous effectuons un tirage aléatoire afin de constituer les matrices d'apprentissage γ_1 et de test γ_2. Nous effectuons ensuite un apprentissage non supervisé avec *k*-means sur la matrice γ_1 qui fournit la classe prédite pour chaque courriel. La dernière étape consiste à présenter γ_1 à la machine à support vectoriel, celle-ci pouvant dès lors effectuer un apprentissage supervisé à l'aide des étiquettes fournis par *k*-means. La généralisation est effectué par la machine SVM sur l'ensemble γ_2 a partir des vecteurs support trouvés précédemment. Ainsi, plusieurs tests indépendants ont été effectués afin d'obtenir des statistiques. La figure 4 montre la chaîne de traitement au complet.

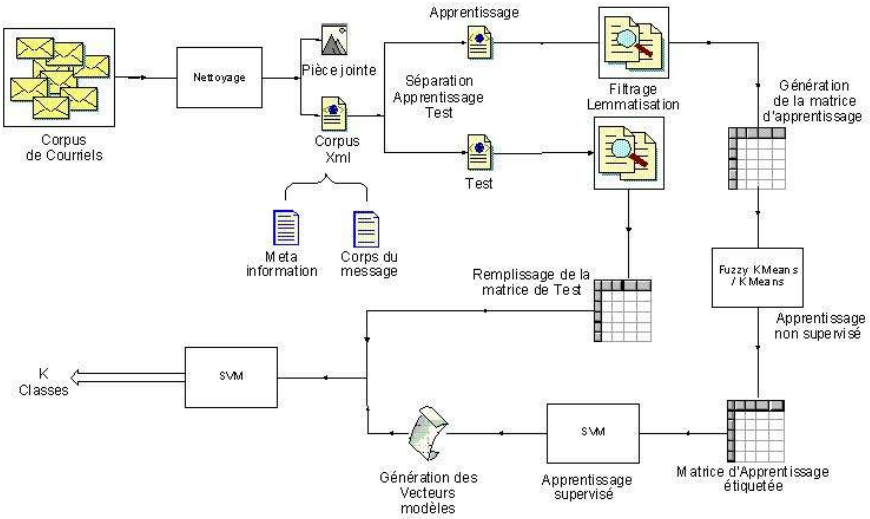
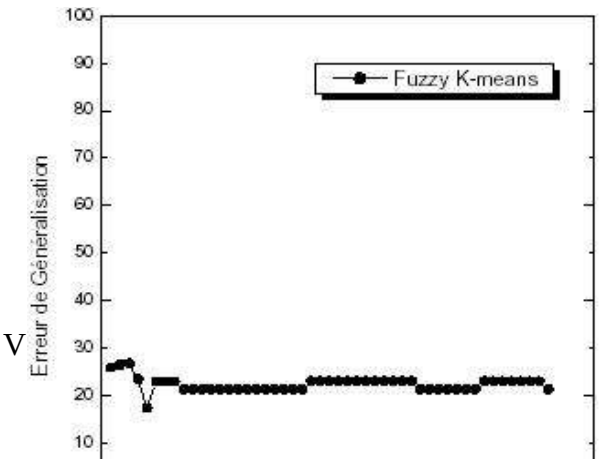


Figure 4 Méthode hybride: chaîne de traitement complète

4 Expériences

Nous avons travaillé avec des corpus de P={200, 500, 1000} courriels ayant k=4 classes parmi {football, jeux de rôles, cinéma, ornithologie}. Chacun des tests a été effectué à 10 reprises aléatoirement. La figure 5 présente l'incidence du paramètre de flou f sur des corpus de P=200 courriels. La figure 6 présente une comparaison entre une initialisation aléatoire et une initialisation semi supervisée où l'on prend une petite partie P_{ini} de l'ensemble d'apprentissage γ_1 (ici $P_{ini} = 0.2P$) afin d'avoir des centroïdes initiaux de meilleure qualité. Nous constatons une diminution dramatique de l'erreur d'apprentissage ε_a avec l'initialisation semi supervisée.

La figure 7 présente les résultats obtenus sur un corpus P=500. A gauche nous montrons l'erreur d'apprentissage ε_a avec une initialisation semi supervisée pour *k*-means. A droite, nous montrons l'erreur en généralisation ε_g de SVM avec un apprentissage supervisé. Bien sûr, l'erreur est faible mais la [illegible] exemples de l'ensemble d'apprentissage.



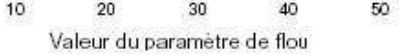


Figure 5 Incidence du paramètre flou sur la généralisation

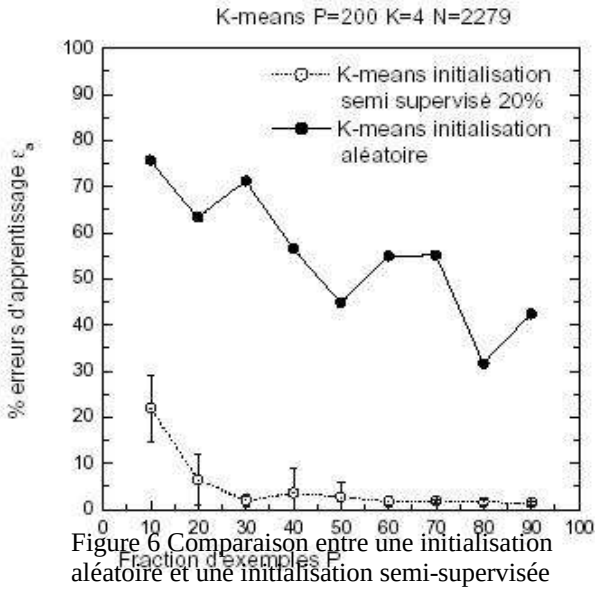


Figure 6 Comparaison entre une initialisation aléatoire et une initialisation semi-supervisée

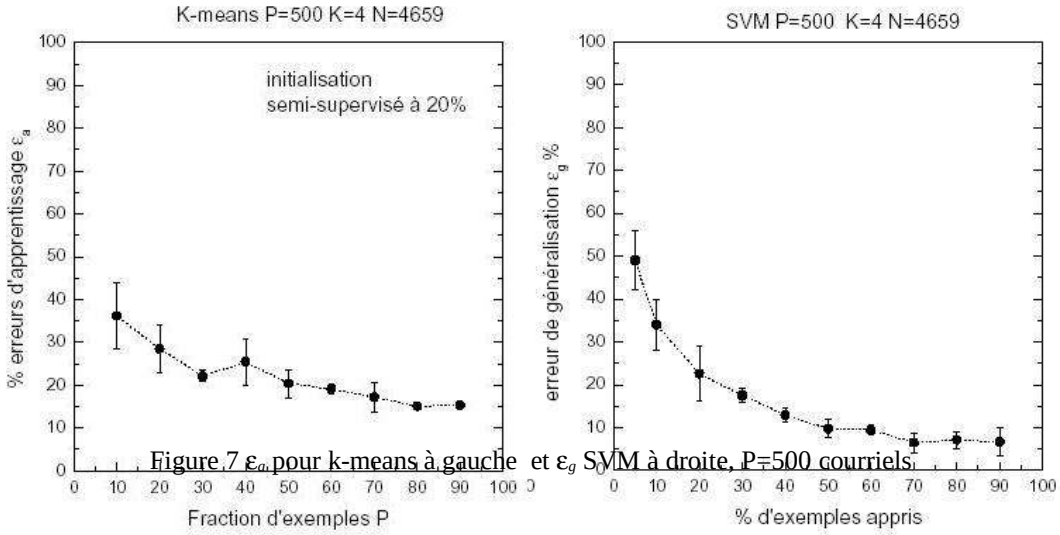


Figure 7 ε_a pour k-means à gauche et ε_g SVM à droite, P=500 courriels

Les figures 8 et 9 comparent les résultats de la méthode hybride et des SVM supervisées sur des corpus de P={200, 500, 1000} courriels. Dans le cas hybride, nous avons combiné un apprentissage non supervisée par *k*-means (avec une initialisation semi supervisée de 0.2P courriels) et supervisée par SVM. Nous constatons que la performance ne se détériore pas en augmentant la taille du corpus. On voit aussi que les performances en généralisation de la méthode hybride sont très proches de celles des SVM supervisées.

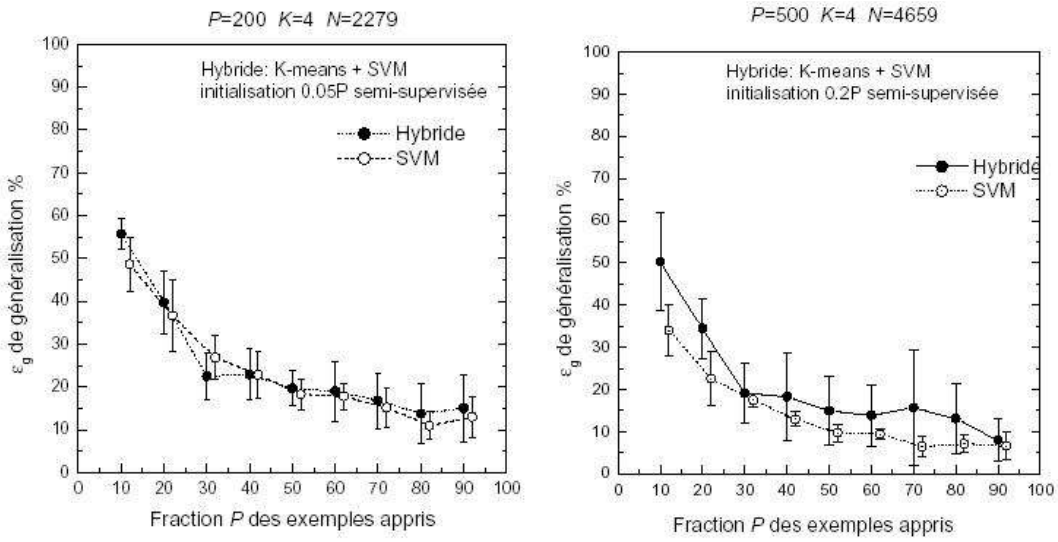


Figure 8 Méthode hybride vs. SVM, *P*=200 courriels à gauche et *P*=500 courriels à droite.

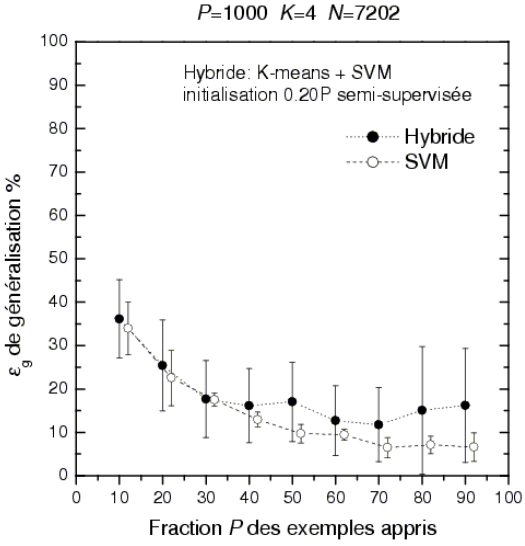


Figure 9 Méthode hybride vs. SVM, *P*=1000 courriels

5 Conclusion et perspectives

La tâche de classification de courriels est assez difficile en raison des particularités de cette forme de communication. Il s'agit d'une tâche où l'on travaille avec des événements rares. Nous avons effectué des processus de prétraitement (filtrage, traduction, lemmatisation) afin de représenter les courriels

dans un modèle vectoriel[16]. A partir de cette représentation des données, nous avons réalisé une étude de différentes méthodes d'apprentissage automatique. Cette étude nous a permis de mieux connaître les caractéristiques de ces méthodes et leur comportement sur nos données. Ainsi, l'apprentissage supervisé permet de mieux classer des nouveaux courriels mais demande une classification préalable qui n'est pas toujours facile à mettre en œuvre. Par contre, bien que l'erreur d'apprentissage de *k*-means et de fuzzy *k*-means avec initialisation aléatoire semble importante, nous l'avons diminué avec une initialisation semi supervisée ayant un faible nombre d'exemples. De plus, cette méthode n'a pas besoin de données étiquetées. La méthode hybride, qui permet de combiner les avantages de l'apprentissage non supervisé de *k*-means pour pré-étiqueter les données, et du supervisé avec SVM pour trouver les séparateurs optimaux, a donné des résultats intéressants. Nous nous sommes principalement intéressés à l'amélioration de l'apprentissage non supervisé, celui-ci ayant les résultats les plus bas au départ. Nos résultats montrant que la performance du système hybride est proche de celle des SVM. La prochaine étape consistera à optimiser les SVM afin de pouvoir améliorer les performances globales du système. Ainsi, une implémentation des machines à support vectoriel selon l'algorithme DDAG [3] permettrait d'éliminer les régions inclassifiables et obtenir une meilleure classification des nouvelles données. De même, une combinaison de classifieurs [7] pourrait permettre d'améliorer nos résultats. Nous prévoyons par la suite aussi d'augmenter la taille des corpus ainsi que le nombre de classes.

Remerciements

Nous tenons particulièrement à remercier Patrice Bellot, Teva Merlin, Karine Lavenus et Grégoire Moreau ainsi que tous les membres de l'équipe TALNE du LIA.

6 Bibliographie

[1] BEAUREGARD S. ,*Génération de texte dans le cadre d'un système de réponse automatique à des courriels*, Mémoire de maîtrise Université de Montréal, Québec, Canada, 2001.

[2] BEZDEK, J.C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

[3] BOONSERM, NITIWUT ,*Multiclass Support Vector Machines Using Adaptive Directed Acyclic Graph*, http://bioinfo.cpgei.cefetpr.br/anais/WCCI02/IJCNN02/PDFFiles/Papers/1263.pdf

[4] COLLOBERT R. & BENGIO S. ,*On the Convergence of SVMTorch, an algorithm for Large-Scale Regression problem*, http://www.ai.mit.edu/projects/jmlr/pa pers/volume1/collobert01a/collobert01a.pdf 2000.

[5] DEGRUIJTER, J.J., MCBRATNEY, A.B., ,*A modified fuzzy k means for predictive classification*.

[6] FLEMM, *Un analyseur flexionnel du français à base de règles*, Fiammetta Namer, TAL vol. 41 No. 2/2000 pp 523-247.

[7] GRILHERES, B., BRUNESSAUX S. and LERAY P., *Combining classifiers for harmful document filtering*, RIAO 2004, pp. 173-185.

[8] KESSLER R., TORRES-MORENO J.M., EL-BEZE M., *Classification thématique de courriels*, Journée sur les nouvelles formes de communication écrite, juin 2004 ATALA Paris.

[9] KOSSEIM L. et LAPALME G., *Critères de sélection d'une approche pour le suivi automatique du courriel*, Actes de la 8ème conférence annuelle sur le Traitement Automatique des Langues Naturelles (TALN-2001), pp. 357-371, juillet 2001, Tours, France.

[10] MANNING D.C. & SCHÜTZE H., *Foundations of Statistical Natural Language Processing. The MIT Press, 2000*.

[11] TORRES-MORENO J.M., BOUGRAIN L. and ALEXANDRE F., *Database Classification by Hybrid Method combining Supervised and Unsupervised Learnings*, 2003 ICANN/ICONIP 2003, pp 37-40.

[16] Nous travaillons actuellement sur d'autres heuristiques de prétraitement afin d'améliorer le filtrage et la lemmatisation, telles que la suppression des accents dans les dictionnaires, détection de fautes par ajout/supression de lettres... Des premiers tests ont montré un gain de 5% de lemmatisation sur des mots qui à la base étaient mal orthographiés.

[12] TORRES-MORENO, J.M., VELAZQUEZ-MORALES, P. et MEUNIER, J.G., *Condensés de textes par des méthodes numériques*, JADT 2002, V2:723-734, A. Morin & P. Sébillot éd, IRISA, France 2002.

[13] TORRES-MORENO, J.M., VELAZQUEZ-MORALES, P. et MEUNIER, J.G., *Cortex : un algorithme pour la condensation automatique des textes*, ARCo 2001, La cognition entre individu et société ARCo 2001. Hermès Science France. pp 365 + vol 2. ISC-Lyon, pp 65-5, Décembre 2001.

[14] TORRES-MORENO, J.M., VELAZQUEZ-MORALES, P. et MEUNIER, J.G., *Classphères : un réseau incrémental pour l'apprentissage non supervisé appliqué à la classification de textes*, JADT 2000, pp 365-372, M. Rajman & J.-C. Chappelier éditeurs, EPFL, 2000.

[15] VAPNIK V., *The Nature of statistical Learning Theory (second ed.)*, Springer, 1995.

[16] VINOT R., GRABAR N., Valette M., *Application d'algorithmes de classification automatique pour la détection des contenus racistes sur l'Internet*, http://www.stud.enst.fr/~vinot/publi/ taln2003.pdf 2003

[17] WARTEL D., *Algorithmes de clustering*, www.galilei.ulb.ac.be/rd/priv_publi/ClusteringAlgorithms.pdf 2003.