

Un nouveau système indépendant de rejet multi-seuils pour la reconnaissance de mots manuscrits

Laurent Guichard, Alejandro Toselli, Bertrand Coüasnon

► **To cite this version:**

Laurent Guichard, Alejandro Toselli, Bertrand Coüasnon. Un nouveau système indépendant de rejet multi-seuils pour la reconnaissance de mots manuscrits. RFIA, Jan 2010, Caen, France. pp.471-478. <hal-01197490>

HAL Id: hal-01197490

<https://hal.inria.fr/hal-01197490>

Submitted on 11 Sep 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Un nouveau système indépendant de rejet multi-seuils pour la reconnaissance de mots manuscrits

Laurent Guichard¹

Alejandro H. Toselli¹

Bertrand Couasnon²

¹ INRIA Rennes - Bretagne Atlantique, Campus de Beaulieu, F-35042 Rennes

² INSA de Rennes, Avenue des Buttes de Coësmes, F-35043 Rennes

UMR IRISA, Campus de Beaulieu, F-35042 Rennes
Université Européenne de Bretagne, France
{lguichar,atoselli,couasnon}@irisa.fr

Résumé

En reconnaissance de mots manuscrits, la capacité de rejeter les mots qui n'appartiennent pas au lexique ou présentent une ambiguïté est indispensable pour fiabiliser un système de reconnaissance utilisé en condition réelle.

Dans cet article, un nouveau système de rejet reposant sur des classifieurs de type séparateur à vaste marge (SVM) et sur un algorithme original d'apprentissage de seuils multiples est présenté.

En substance, nous proposons d'utiliser un ensemble de SVM pour réévaluer les hypothèses produites par un système de reconnaissance de mots classique à base de modèles de Markov cachés. Le mécanisme de vérification rejette ou accepte ces hypothèses en comparant leur mesure de confiance aux seuils de rejet. Un algorithme optimal de programmation dynamique qui vise à apprendre ces seuils de rejet en maximisant le taux de reconnaissance pour un taux d'erreur désiré est également détaillé.

Les expérimentations ont été menées sur la base nationale RIMES du programme Techno-Vision et les résultats présentés montrent la supériorité de cette approche sur les autres méthodes existantes en fiabilité et en performance.

Mots Clef

Système de vérification, apprentissage de seuils de rejet, mesure de confiance par SVM, reconnaissance de mots manuscrits

Abstract

In the field of isolated handwritten word recognition, the development of highly effective verification systems to reject words that do not belong to the lexicon or present ambiguities is still an active research topic.

In this paper, a novel verification architecture based on support vector machine scoring and on a new algorithm for tuning multiple reject class-dependent thresholds is presented. In essence, a set of support vector machines appended to a standard HMM based recognition system provides

class-dependant confidence measures employed by the verification mechanism to accept or reject the recognized hypotheses. In order to tune these class-dependent rejection thresholds, an algorithm based on dynamic programming is proposed which focus on maximizing the recognition rate for a given imposed error rate.

Preliminary reported results of experiments carried out on Techno-Vision program RIMES database show that this approach outperforms state of the art rejection methods.

Keywords

Verification system, multiple thresholds tuning, SVM-based confidence measure, handwriting word recognition

1 Introduction

Dans le contexte de la reconnaissance de mots manuscrits (*Handwritten Word Recognition*, HWR), l'intégration de systèmes de vérification (*Verification Systems*, VS) efficaces est particulièrement importante. Ces VS doivent détecter le manque de fiabilité dans les hypothèses de sortie du HWR et ainsi permettre de rejeter le mot en entrée. De tels systèmes de reconnaissance avec rejet (HWR-VS) sont cruciaux et vitaux pour de nombreuses applications industrielles comme l'analyse automatique de courriers manuscrits entrants d'une entreprise, la reconnaissance de plaques d'immatriculation [11], de montant de chèques [18]...

De manière classique, les VS sont constitués de deux étapes : le calcul de la mesure de confiance qui donne une information sur la qualité de la reconnaissance et l'application de seuils pour minimiser le rejet et l'erreur, afin de fiabiliser la reconnaissance.

Nous proposons dans cet article un VS indépendant du HWR, c'est-à-dire pouvant être mis en place sans modifier et refaire un apprentissage du HWR. Ce dernier doit être capable de proposer les n-meilleures réponses ainsi que les segmentations en graphèmes des mots correspondants. En s'appuyant sur ces seuls éléments, le VS que nous proposons offre deux principales contributions :

- une nouvelle méthode de vérification qui réévalue les hypothèses du HWR grâce à des SVM et effectue un rejet multi-seuils sur la meilleure d’entre elles,
- un nouvel algorithme qui, contrairement aux autres approches, garantit un apprentissage optimal de ces seuils de rejet.

Après une étude des VS existants dans la littérature, nous présentons dans la section 3 notre VS et ces deux contributions principales. La section 4 décrit ensuite le HWR que nous avons utilisé pour valider le VS. Cette validation s’appuie sur des expérimentations effectuées sur la base RIMES [5] (programme national Techno-Vision), et présentées dans la section 5. Cette section présente et discute également les résultats obtenus en comparaison avec d’autres méthodes.

2 État de l’art

La littérature décrit une grande variété de HWR-VS qui sont principalement dépendants de la nature intrinsèque du HWR à vérifier [9, 14, 12, 8]. Si nous considérons les HWR fondées sur des Modèles de Markov Cachés (*Hidden Markov Models*, HMM), où la probabilité d’émission de chaque état est modélisée par des mixtures de gaussiennes (*Gaussian mixture model*, GMM), les systèmes de validation proposés s’appuient sur les mesures de confiance issues du décodage des HMM comme par exemple dans [9]. D’autres approches sont fondées sur la réévaluation des hypothèses par des perceptrons multi-couches (*Multi-Layer Perceptron*, MLP) comme dans [10]. Bien que les classificateurs de type SVM aient montré leur bonne capacité de rejet pour la reconnaissance de caractères isolés en-ligne [17] ou celle de chiffres manuscrits [2], ils n’ont pas été employés à notre connaissance dans le domaine de la reconnaissance de mots manuscrits mais uniquement dans celui du traitement de la parole [13].

Comme indiqué précédemment, les approches des systèmes de vérification s’appuient sur des méthodes de seuillage qui nécessitent des ajustements (manuels ou automatiques) des seuils pour décider d’accepter ou de rejeter les hypothèses de reconnaissance. Le nombre de seuils utilisés pour accepter ou rejeter une hypothèse donnée est variable. On trouve ainsi des VS utilisant un seuil unique ou plusieurs.

[3] propose le meilleur compromis erreur-rejet ainsi que la règle optimale associée de rejet. Selon cet article, le compromis optimal erreur-rejet peut être atteint uniquement si les probabilités *a posteriori* sont connues exactement, ce qui n’est pas possible dans des expérimentations réelles. Cependant, [4] suggère l’utilisation de seuils multiples de rejet pour obtenir des régions optimales de décision de rejet, même dans le cas où les probabilités *a posteriori* sont affectées par des erreurs. En outre, ces travaux, ainsi que d’autres comme [7, 9, 16], proposent différents algorithmes pour trouver des valeurs adéquates pour les seuils de rejet, mais aucun d’entre eux ne garantit une version optimale.

3 Système de vérification

Le VS présenté dans cet article est indépendant du HWR auquel il est ajouté. Cependant, ce dernier doit produire pour un exemple qui lui est soumis une liste des N meilleures hypothèses avec leur segmentation en graphèmes. Les HWR de type segmentation/reconnaissance ou à base de HMM répondent à cette exigence. En sortie, le VS donne une sanction : soit l’exemple est rejeté, soit une des hypothèses du HWR est acceptée. Ce VS est constitué d’une réévaluation des hypothèses du HWR par des SVM, d’un rejet multi-seuils et d’un algorithme de programmation dynamique pour l’apprentissage de ces seuils. Ces différents éléments sont détaillés ci-dessous.

3.1 Méthode de vérification

Comme le montre le diagramme global de la figure 1, le VS prend en entrée une liste des N meilleures hypothèses H_i avec leur segmentation en graphèmes :

$$H_i : (S_i, W_i)$$

avec $i \in [0, N - 1]$, N le nombre d’hypothèses dans la liste. S_i , la segmentation issue du HWR, est constituée d’une séquence de graphèmes $\langle g_{i,0}, \dots, g_{i,n_i-1} \rangle$ et W_i , le mot du lexique attribué par le HWR, est la concaténation de classes de graphème $\langle c_{i,0} \dots c_{i,n_i-1} \rangle$ où n_i est le nombre de graphèmes dans l’hypothèse H_i . La paire $(g_{i,j}, c_{i,j})$ désigne donc le graphème $g_{i,j}$ et la classe $c_{i,j}$ qui lui est associée.

La première étape du VS utilise la segmentation en graphèmes pour découper l’image d’entrée en plusieurs sous-images, chacune représentant un caractère (= graphème) dans notre exemple. Ce principe se fonde sur l’hypothèse que la segmentation donnée par le système de HWR est fiable et que donc une sous-image correspond à un graphème. Ensuite, une fonction d’extraction transforme chaque sous-image en un vecteur de 95 caractéristiques $v_{i,j}$ composés :

- des moments de Zernike d’ordre 8 : 45 dimensions
- des histogrammes des directions des contours codés en chaîne de Freeman : 48 dimensions
- de la répartition des pixels du graphème dans la zone haute et moyenne : 2 dimensions. Zones définies par rapport à ligne haute et basse du corps du mot.

La deuxième étape réévalue chaque graphème à l’aide d’un ensemble de classificateurs SVM. Le nombre de classificateurs est équivalent au nombre de classes de graphèmes différentes présentes dans les mots du lexique. Pour une paire donnée $(g_{i,j}, c_{i,j})$, le SVM correspondant à la classe $c_{i,j}$ prend en entrée le vecteur de caractéristiques $v_{i,j}$ extrait à la première étape du VS et lui attribue un nouveau score $P(c_{i,k} | g_{i,k})$.

Pour convertir les scores des SVM en probabilités, une fonction *softmax* leur est appliquée comme décrit dans [15].

A présent qu’une probabilité a été affectée à chaque graphème de chaque hypothèse, une mesure de confiance glo-

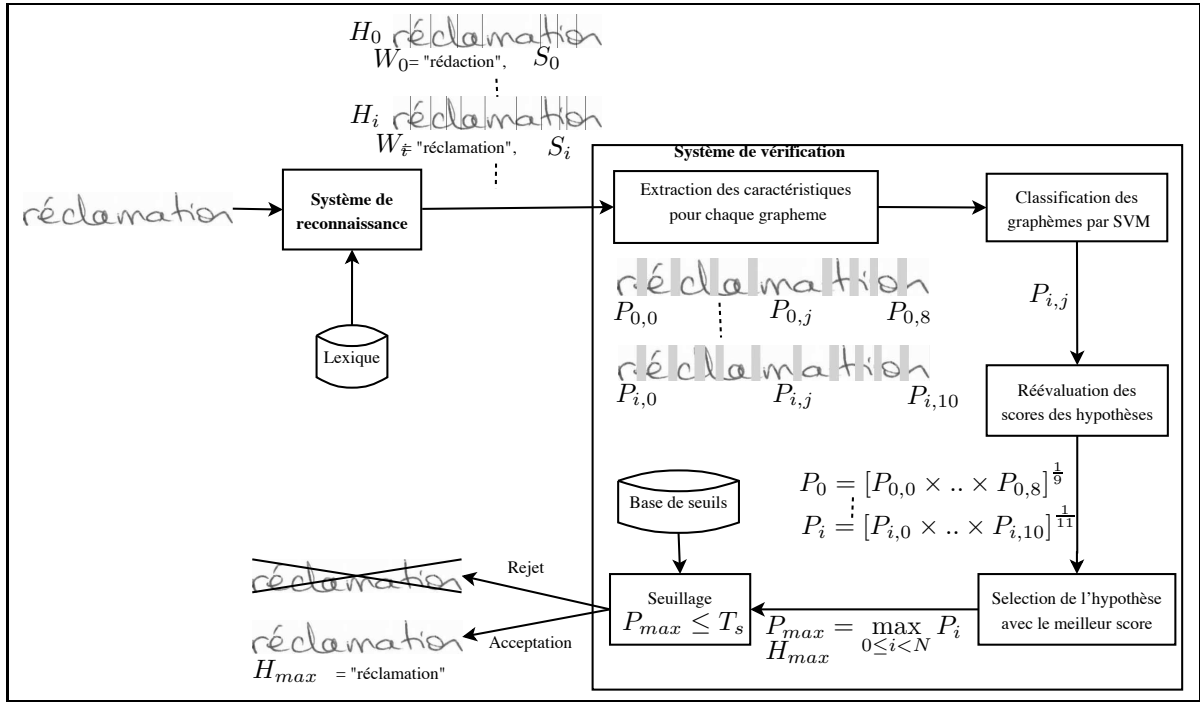


FIG. 1: Diagramme global du système de vérification

bale $P(H_i)$ est calculée pour chaque hypothèse en faisant la moyenne géométrique des probabilités des graphèmes :

$$P_i = P(H_i) = \left[\prod_{k=0}^{n_i-1} P(c_{i,k} | g_{i,k}) \right]^{\frac{1}{n_i}}$$

En théorie, la mesure de confiance devrait être le produit des probabilités des graphèmes mais l'expérience a montré que normaliser par la racine n-ième donne de meilleurs résultats. Ceci s'explique par le fait que les hypothèses peuvent avoir des longueurs (i.e. nombre de graphèmes) différentes et que l'application d'une moyenne géométrique rend donc les mesures de confiance comparables. A l'issue de cette étape de réévaluation, chaque hypothèse dispose d'une nouvelle mesure de confiance.

Pour l'étape de décision, i.e. acceptation ou rejet de l'exemple en entrée, on considère uniquement l'hypothèse avec la mesure de confiance maximale :

$$P_{max} = \max_{i=0, \dots, N-1} P_i$$

Ce choix de ne garder que l'hypothèse avec la meilleure mesure de confiance est fait pour se conformer à la règle définie dans [4], extension multi-seuils de la règle de rejet optimal de Chow [3], pour rester comparable aux autres méthodes existantes et enfin parce qu'il est requis par les différents algorithmes d'apprentissage de seuils.

Comme évoqué précédemment, le mécanisme de seuillage utilise plusieurs seuils, chacun correspondant à une classe. On définit l'ensemble des *classes de longueur de mot* Ω

par :

$$\Omega = \{length(W) : W \in Lex\}$$

où Lex est le lexique et $length$ est une fonction qui renvoie le nombre de graphèmes dans le mot W . Cela implique que le nombre de *classes de longueur de mot* N_c vaut $|\Omega|$. Ainsi, la *classe de longueur* du mot W_i de l'hypothèse H_i se calcule par :

$$C_j = length(W_i) = n_i$$

avec $j \in [0, N_c - 1]$.

Cette technique d'utilisation de *classes de longueur de mot* qui regroupe les mots de la même longueur dans une classe est motivée par l'intuition que l'acceptation d'un mot de 10 caractères contenant 1 caractère dégradé est différente (i.e. moins risquée) de l'acceptation d'un mot de 2 caractères contenant aussi 1 caractère dégradé.

Chaque classe C_j est associée à un seuil de rejet T_j . Au moment de la vérification, le seuil $T_{C_{max}}$ correspondant à la classe C_{max} de l'hypothèse H_{max} est sélectionné et la règle suivante est appliquée :

- Si $P_{max} < T_{C_{max}}$ Alors l'exemple est rejeté.
- Si $P_{max} \geq T_{C_{max}}$ Alors l'exemple est accepté.

Notre mécanisme de décision se déroule donc en deux temps : d'abord sélection de la meilleure hypothèse suivant sa mesure de confiance (inter-classe) et ensuite acceptation/rejet de l'hypothèse retenue par comparaison de sa mesure de confiance au seuil de rejet adéquat (intra-classe).

3.2 Algorithme d'apprentissage de seuils

La fiabilité du VS présenté ici repose sur un ensemble de seuils. Le réglage des seuils peut être vu comme un pro-

blème d'optimisation.

Soit une base contenant N_{total} exemples et soit :

$$\begin{aligned} \text{Performance} &= Perf = \frac{R}{N_{total}} \\ \text{Taux d'erreur} &= ER = \frac{E}{N_{total}} \end{aligned} \quad (1)$$

où E (resp. R) est le nombre d'exemples mal (resp. bien) classés.

A partir de [4], le problème de réglage des seuils peut être formulé de la façon suivante :

$$\begin{cases} \max_{T_0, \dots, T_{N_c-1}} Perf(T_0, \dots, T_{N_c-1}) \\ ER(T_0, \dots, T_{N_c-1}) \leq ER_{max} \end{cases} \quad (2)$$

avec ER_{max} le taux d'erreur maximal autorisé.

L'idée est de trouver les seuils qui maximisent la performance sans excéder un certain taux d'erreur. Cette formulation a été choisie parce qu'elle est communément utilisée dans les applications réelles et qu'elle est facilement interprétable.

Comme évoqué précédemment, les algorithmes existants pour le réglage de seuils multiples ne sont pas optimaux [4], [16] et [7]. L'algorithme présenté ici est inspiré de la résolution du problème du sac à dos binaire. Ce problème consiste à choisir parmi un ensemble d'objets, chacun affublé d'un poids et d'une valeur, ceux dont le poids total ne dépasse pas la charge limite du sac à dos tout en maximisant la valeur totale.

La solution par programmation dynamique au problème du sac à dos binaire a été modifiée pour résoudre le problème de réglage des seuils.

D'abord, (2) est récrit dans une formulation absolue :

$$\begin{cases} \max_{T_0, \dots, T_{N_c-1}} R(T_0, \dots, T_{N_c-1}) \\ E(T_0, \dots, T_{N_c-1}) \leq E_{max} \end{cases} \quad (3)$$

Pour régler les seuils, une base d'apprentissage est requise. Elle est constituée de N_{total} exemples. A chaque exemple s de la base est associé un quadruplet (P_s, C_s, E_s, R_s) : P_s et C_s sont la probabilité et la classe affectées par le VS à cet exemple. E_s (resp. R_s) correspond au nombre d'exemples qui seraient mal (resp. bien) classé par le VS si P_s était choisie comme seuil de rejet pour cette classe.

Ensuite, on introduit $A(C, E)$, le nombre maximal d'exemples bien classés qui peut être atteint avec un nombre d'erreurs inférieur ou égal à E en considérant les classes 0 à C . $A(C, E)$ est défini récursivement par :

$$\begin{cases} A(0, E) = \max_{s \in C=0, E_s \leq E} R_s \\ A(C, E) = \max_{s \in C, E_s \leq E} A(C-1, E-E_s) + R_s \end{cases}$$

Cette définition récursive met en évidence la propriété de *sous-structure optimale* de notre problème. En effet, la solution optimale au problème à C classes se construit à partir de celle du problème à $C-1$ classes. Comme dans le cas du sac à dos, un algorithme de programmation dynamique permet de calculer la solution optimale au problème du réglage des seuils.

Avant de lancer l'apprentissage, pour éviter les problèmes liés à une classe vide, un exemple défini par $(1.0, C, 0, 0)$ est ajouté à chaque classe. Pendant l'apprentissage, l'exemple qui maximise $A(C, E)$ à chaque pas est stocké dans $S(C, E)$. Le calcul de $A(C, E)$ est réalisé jusqu'à ce que $C = N_c - 1$ et $E = E_{max}$.

Enfin, les seuils de rejet sont récupérés par retour arrière comme l'explique l'algorithme 1.

Algorithm 1 Récupération des seuils de rejet

```

T : Tableau des seuils à récupérer
// Initialisation
C ← Nc - 1 {Nc : nombre de classes}
E ← Emax {Emax : erreur maximale autorisée}
// Récupération des seuils
while C ≥ 0 do
  s ← S(C, E)
  T[C] ← Ps {Seuil pour la classe C}
  E ← E - Es
  C ← C - 1 {Passe à la classe suivante}
end while

```

4 Description de l'application de HWR

Pour tester le VS, nous utilisons un système de reconnaissance de mots manuscrits basé sur les modèles de Markov cachés. Il est composé de 3 modules : pré-traitement, extraction des vecteurs de caractéristiques et reconnaissance. Le module de pré-traitement, adapté aux approches à base de HMM et ayant pour but la correction/normalisation de l'exemple en entrée, est constitué des éléments suivants : suppression du bruit, correction de l'inclinaison et de la pente. Toutes ces étapes utilisent des techniques de l'état de l'art.

Le module d'extraction des caractéristiques transforme par un mécanisme de fenêtre glissante l'image du mot en entrée en une séquence de vecteurs de caractéristiques de dimensions 60. La méthode d'extraction des caractéristiques est décrite dans [1].

Comme mentionné précédemment, le processus de reconnaissance est basé sur des HMM, où les caractères (l'unité élémentaire de la reconnaissance) sont modélisés par des HMM gauche-droite ayant un nombre variable d'états. Plus précisément, le nombre d'états pour un HMM donné est estimé à partir de la moyenne des longueurs des séquences de vecteurs utilisés pour l'apprentissage. De plus, les probabilités d'émission d'un vecteur par état sont modélisées par une mixture de Gaussiennes.

Le nombre optimal d'états par HMM ainsi que le nombre de Gaussiennes par état (128 dans notre cas) ont été réglés à partir de la base de validation RIMES. L'algorithme de Baum-Welch a été employé pour estimer les paramètres des modèles de caractères (HMM) sur la base d'apprentissage et de validation RIMES.

Dans la phase de décodage, chaque mot du dictionnaire est représenté par la concaténation des modèles de caractère et l'algorithme de Viterbi est utilisé pour rechercher la concaténation qui maximise la probabilité d'émission de la séquence de vecteurs de caractéristiques en question.

5 Expérimentations

Afin de tester l'efficacité du VS présenté, plusieurs expérimentations ont été réalisées. La base de données RIMES, les mesures de performance et les résultats sont expliqués ci-dessous.

5.1 Base de données RIMES-DB

La base de données RIMES [6]¹ (Reconnaissance et Indexation de données Manuscrites et de fac-similés) a été utilisée.

Cette base est composée de plus de 12 000 pages (lettres, courriels, faxes et formulaires) entièrement annotées, de 100 000 caractères segmentés, de 300 000 mots isolés et de 500 logos. Pour notre problème, nous nous intéressons uniquement aux images de mot manuscrites extraites de lettres numérisées à 300 dpi et stockées au format PNG. Cette base de mots multi-scripteurs est caractérisée par des styles, des polices et des tailles d'écriture différents. La figure 2 présente des exemples d'images de mot.

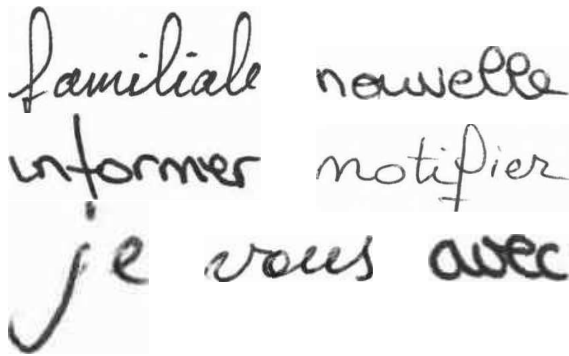


FIG. 2: Exemples d'images de mot de la base RIMES.

La base contient donc 59 202 mots avec leur annotation ce qui forme un lexique de 1 612 mots. Le tableau 1 résume ces informations par partition.

5.2 Méthodes d'évaluation

Pour évaluer les résultats du VS proposé, les mesures suivantes sont utilisées :

Le taux de bon rejet (*true rejection rate*, TRR) :

¹Projet RIMES (<http://rimes.it-sudparis.eu>), programme *Techno-Vision* soutenu par les Ministères de la Recherche et de la Défense (France).

TAB. 1: Statistiques sur les partitions de la base RIMES.

Nombre de :	App.	Valid.	Test	Total	Lex.
mots	44 196	7 542	7 464	59 202	1 612
caractères	230 259	39 174	38 906	308 339	65

$$\text{TRR} = \frac{\text{\#mots rejetés à raison}}{\text{\#mots à rejeter}}$$

Le taux de faux rejet (*false rejection rate*, FRR) :

$$\text{FRR} = \frac{\text{\#mots rejetés à tort}}{\text{\#mots à accepter}}$$

La performance et le taux d'erreur ont été définis en (1).

Les courbes ROC (*Receiver Operating Characteristic*) qui représentent le TRR en fonction du FRR sont utilisées. L'aire sous une courbe ROC (AROC) donne une bonne estimation de la capacité de rejet d'un système de vérification, c'est à dire sa capacité à rejeter les exemples mal classés sans rejeter d'exemples bien classés. Une aire sous la courbe de 1.0 indique que le système peut rejeter tous les exemples qui doivent l'être sans en rejeter à tort. L'emploi de la courbe ROC permet d'évaluer et de comparer notre système de vérification à d'autres approches déjà publiées. La courbe de la performance en fonction de l'erreur est aussi utilisée pour rendre compte du fait qu'un VS améliore également le taux d'exemples bien classés.

5.3 Résultats

Dans cette partie, un certain nombre de comparaisons ont été réalisées entre notre système et d'autres approches. Deux types d'expérimentations ont été menées.

La première, qui teste la méthode, a pour but d'évaluer l'apport des SVM et des seuils multiples. Nous avons comparé notre méthode avec deux autres **SVM-single** et **HMM-dif_12**, détaillées ci-dessous. Les résultats sont présentés dans la figure 3.

La seconde, qui teste l'algorithme, a pour but de confirmer que notre algorithme de programmation dynamique de réglage des seuils fait mieux ou aussi bien que les autres algorithmes, dans le cas présent **CMP** et **AMTL**. Le test de ces algorithmes est réalisé à méthode constante, c'est à dire en utilisant la réévaluation des hypothèses par les SVM et les seuils multiples. Les résultats sont présentés dans le tableau 2 ainsi que dans la figure 4.

La liste suivante présente les différents systèmes utilisés dans nos expérimentations :

SVM-multi-DP : Il s'agit du système que nous proposons décrit dans la partie 3 constitué de la méthode de vérification de la section 3.1 et de l'algorithme de la section 3.2.

SVM-single : Idem **SVM-multi-DP** sans utilisation de *classes de longueur de mot* mais d'un seuil unique global appliqué à la meilleure des hypothèses réévaluées.

HMM-dif_12 : Il ne s'agit pas d'un VS car le rejet est effectué directement sur les scores proposés par le HWR. Dans ce cas, un seuil unique est utilisé qui dépend de la différence entre le 1^{er} et le 2^{ème} meilleur score du HWR. D'après [9], cette méthode donne les meilleurs résultats pour un HWR à base de HMM.

CMP : (*Constrained Maximization Problem*) Idem **SVM-multi-DP** sauf que l'algorithme de réglage des seuils est celui présenté dans [4], qui utilise une procédure itérative.

AMTL : (*Automatic Multiple-Thresholds Learning algorithm*) Idem **SVM-multi-DP** sauf que l'algorithme utilisé pour le réglage des seuils est celui présenté dans [16], qui est proche de celui décrit dans [4] tout en étant plus rapide et plus robuste dans la phase d'initialisation.

Pour l'apprentissage des paramètres du HWR et des SVM du VS, la base d'apprentissage a été employée. Le réglage des seuils a été effectué sur la base de validation. Enfin, la base de test a été utilisée pour réaliser la comparaison des systèmes. Le détail des bases est présenté dans le tableau (1). Sur cette base, le nombre de SVM, égal au nombre de classes de caractères, est de 65. Ces SVM ont été entraînés avec la politique un-contre-tous. Pour ce faire, le système HWR a du être utilisé en mode "alignement forcé" avec un lexique réduit à la vérité terrain. Les paramètres d'apprentissage des SVM : noyaux de type gaussien, constante C = 100 et écart-type = 20.

Pour les systèmes **SVM-multi-DP**, **CMP** et **AMTL**, le nombre de seuils qui correspond au nombre de *classes de longueur de mot* générées par le regroupement des mots du lexique de même longueur est de 17. En effet, le lexique RIMES contient des mots dont la longueur varie de 1 à 17 (toutes les longueurs entre ces bornes étant représentées). Le nombre d'hypothèses produites par le HWR est de 10.

TAB. 2: Capacité de rejet : aire sous la courbe ROC pour les différents algorithmes de réglage de seuils.

Algorithme	AROC	
	Validation	Test
SVM-multi-DP	0.90	0.90
CMP	0.88	0.89
AMTL	0.88	0.89

Résultats de la comparaison des méthodes. A partir des courbes de la figure 3-a, on peut noter que pour un FRR situé dans l'intervalle de 0 à 20% les capacités de rejet d'un HWR (**HMM-dif_12**) sont moins bonnes que celles d'un HWR-VS utilisant la réévaluation des hypothèses par des SVM (**SVM-single**). Dans ce même intervalle, ce dernier système est lui-même surpassé par le HWR-VS employant des seuils multiples (**SVM-multi-DP**). Le tableau 3 donne les TRR pour un FRR de 10%.

De la même façon, les courbes de la figure 3-b montrent que les performances du système HWR-VS multi-seuils

TAB. 3: Comparaison des méthodes : TRR pour un FRR fixé à 10% (lexique de 1612 mots).

Méthode	TRR (%)
SVM-multi-DP	76
SVM-single	73
HMM-dif_12	58

sont meilleures que celles du HWR-VS simple seuil et du HWR seul. La différence est surtout significative dans la première partie de la courbe (pour un taux d'erreur entre 0 et 5%). Au delà, pour des taux d'erreur supérieur à 7%, l'apport des seuils multiples est moins significatif. Le tableau 4 donne les performances pour un taux d'erreur de 2.5%.

TAB. 4: Comparaison des méthodes : performance pour un taux d'erreur fixé à 2.5% (lexique de 1612 mots).

Méthode	Perf. (%)
SVM-multi-DP	61
SVM-single	52
HMM-dif_12	50

Résultats de la comparaison des algorithmes. Pour l'expérimentation sur les algorithmes, les figures 4-a et 4-b montrent que l'algorithme de programmation dynamique est une limite supérieure aux autres algorithmes. On peut également remarquer que les courbes produites par cet algorithme sont monotones ce qui s'explique par sa nature à base de programmation dynamique. Les courbes produites par les autres algorithmes ont des croissances plus erratiques à cause de l'utilisation d'heuristiques. La tableau 2 montre que l'aire sous la courbe ROC de l'algorithme de programmation dynamique est supérieure aux aires des autres algorithmes aussi bien sur la base de validation que sur la base de test ce qui met en évidence sa bonne généralisation.

6 Conclusion

Dans cet article, une méthode alternative de vérification utilisant des SVM pour la réévaluation des hypothèses d'un HWR et des seuils multiples pour le rejet ainsi qu'un nouvel algorithme pour l'apprentissage de ces seuils ont été présentés. Ce VS a comme caractéristique d'être indépendant du HWR auquel il est rattaché dans la mesure où aucune modification de ce dernier n'est nécessaire. Il peut s'utiliser avec des reconnaissances de mots manuscrits de type segmentation/reconnaissance ou à base de HMM.

Dans la section des résultats, nous avons montré que l'emploi des SVM pour faire du rejet est justifié et que l'utilisation de seuils multiples améliore sensiblement les capacités de rejet. De plus, les performances globales du systèmes HWR-VS sont supérieures à celles du HWR seul.

Nous avons par ailleurs mis en évidence l'apport de notre

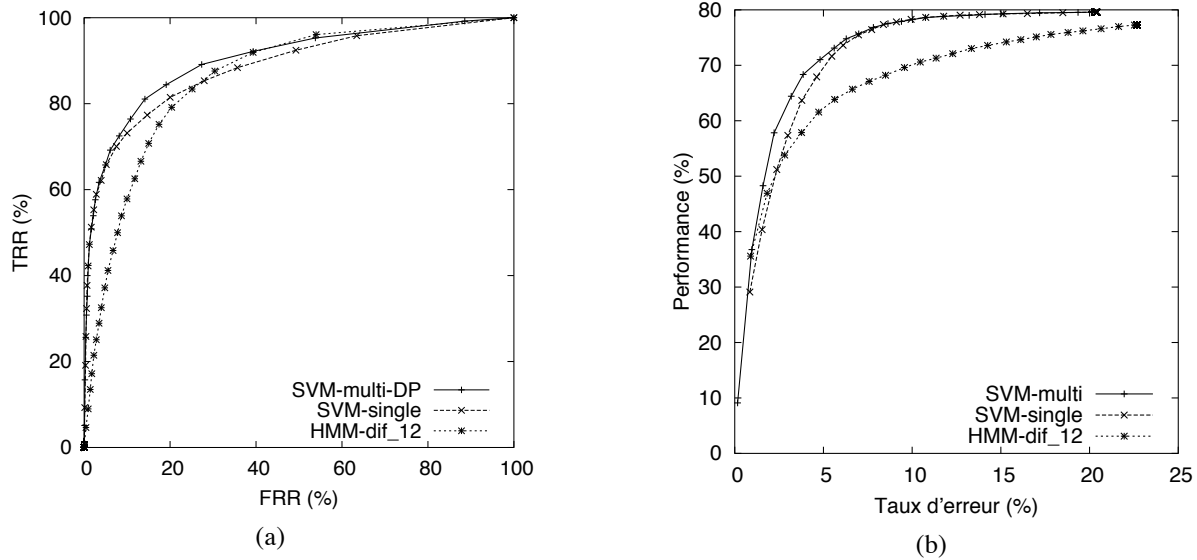


FIG. 3: Résultats des comparaisons des méthodes, réalisées sur la base de test avec un lexique de 1612 mots. A gauche, la courbe ROC et à droite la courbe de la performance en fonction du taux d'erreur.

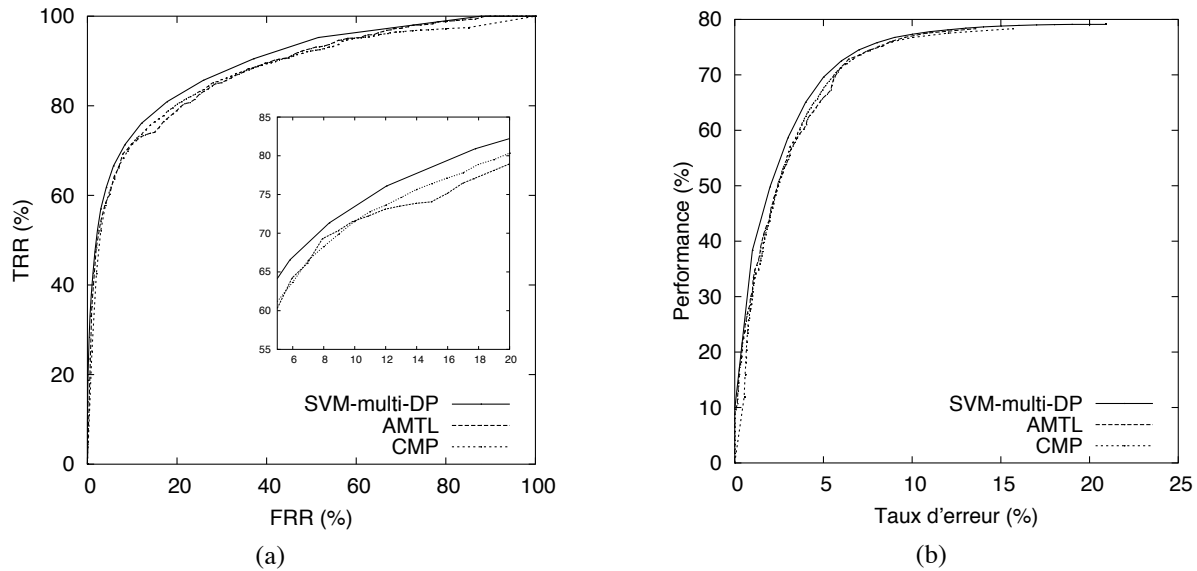


FIG. 4: Résultats des comparaisons des algorithmes, réalisées sur la base de validation avec un lexique de 1612 mots. A gauche, la courbe ROC et à droite la courbe de la performance en fonction du taux d'erreur.

algorithme de programmation dynamique pour régler des seuils multiples. Sa facilité d'emploi (aucun paramètre à choisir), sa rapidité d'exécution et son optimalité font que nous recommandons son usage.

Références

- [1] I. Bazzi, R. Schwartz, and J. Makhoul. An omnifont open-vocabulary ocr system for english and arabic. *IEEE Trans. on PAMI*, 21(6) :495–504, 1999.
- [2] C. Chatelain, L. Heutte, and T. Paquet. A two-stage outlier rejection strategy for numerical field extraction in handwritten documents. In *ICPR, Hong Kong, China*, volume 3, pages 224–227, 2006.
- [3] C. K. Chow. On optimum error and reject tradeoff. *IEEE Information Theory Society*, 16(1) :41–46, January 1970.
- [4] Giorgio Fumera, Fabio Roli, and Giorgio Giacinto. Reject option with multiple thresholds. *Pattern Recognition*, 33 :2099–2101, 2000.
- [5] E. Grosicki and H. El Abed. Icdar 2009 handwriting recognition competition. In *ICDAR '09*, 2009.
- [6] Emmanuele Grosicki, Matthieu CARRE, Jean-Marie Brodin, and Edouard Geoffrois. RIMES evaluation campaign for handwritten mail processing. In *10th International Conference on Document Analysis and Recognition*, pages 941–945, 2009.

- [7] Marcelo N. Kapp, Cinthia O. de A. Freitas, and Robert Sabourin. Methodology for the design of nn-based month-word recognizers written on brazilian bank checks. *Image and Vision Computing*, 25(1):40–49, 2007. SIBGRAPI.
- [8] Marcelo N. Kapp, Cinthia O. De A. Freitas, and Robert Sabourin. Handwritten brazilian month recognition : An analysis of two nn architectures and a rejection mechanism. *Frontiers in Handwriting Recognition, International Workshop on*, 0:209–214, 2004.
- [9] Alessandro L. Koerich. Rejection strategies for handwritten word recognition. In *IWFHR '04 : Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition*, pages 479–484, Washington, DC, USA, 2004. IEEE Computer Society.
- [10] Alessandro L. Koerich, Robert Sabourin, and Ching Y. Suen. Recognition and verification of unconstrained handwritten words. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1509–1522, 2005.
- [11] Hsi-Jian Lee, Si-Yuan Chen, and Shen-Zheng Wang. Extraction and recognition of license plates of motorcycles and vehicles on highways. In *ICPR '04 : Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4*, pages 356–359, Washington, DC, USA, 2004. IEEE Computer Society.
- [12] Jinhui Liu and Paul D. Gader. Neural networks with enhanced outlier rejection ability for off-line handwritten word recognition. *Pattern Recognition*, 35(10):2061–2071, 2002.
- [13] Changxue Ma, M.A. Randolph, and J. Drish. A support vector machines-based rejection technique for speech recognition. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 1:381–384, 2001.
- [14] Sriganesh Madhvanath, Evelyn Kleinberg, and Venu Govindaraju. Holistic verification of handwritten phrases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12):1344–1356, 1999.
- [15] J. Milgram, Mohamed Cheriet, and R. Sabourin. Estimating accurate multi-class probabilities with support vector machines. In *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on*, volume 3, pages 1906–1911 vol. 3, July-4 Aug. 2005.
- [16] H. Mouchere and E. Anquetil. A unified strategy to deal with different natures of reject. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 792–795, 0-0 2006.
- [17] H. Mouchère. *Etude des mécanismes d'adaptation et de rejet pour l'optimisation de classifieurs : Application à la reconnaissance de l'écriture manuscrite en-ligne*. PhD thesis, Institut National des Sciences Appliquées de Rennes.
- [18] Rafael Palacios and Amar Gupta. A system for processing handwritten bank checks automatically. *Image Vision Comput.*, 26(10):1297–1313, 2008.