



**HAL**  
open science

## Adaptation de modèles de Markov cachés - Application à la reconnaissance de caractères imprimés

Kamel Ait-Mohand, Laurent Heutte, Thierry Paquet, Nicolas Ragot

► **To cite this version:**

Kamel Ait-Mohand, Laurent Heutte, Thierry Paquet, Nicolas Ragot. Adaptation de modèles de Markov cachés - Application à la reconnaissance de caractères imprimés. Colloque International Francophone sur l'Écrit et le Document (CIFED2010), Mar 2010, Sousse, Tunisie. pp.83-94. hal-00490060

**HAL Id: hal-00490060**

**<https://hal.science/hal-00490060>**

Submitted on 7 Jun 2010

**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.

---

# Adaptation de modèles de Markov cachés

## Application à la reconnaissance de caractères imprimés

**Kamel Ait-Mohand\*** — **Laurent Heutte\*** — **Thierry Paquet \***—  
**Nicolas Ragot\*\***

\* Université de Rouen, Laboratoire d'Informatique, de Traitement de l'Information  
et des Systèmes, EA 4108

Avenue de l'Université, 76801 Saint-Étienne-du-Rouvray

kamel.ait-mohand@etu.univ-rouen.fr, {Laurent.Heutte, Thierry.Paquet}@univ-  
rouen.fr

\*\* Université François Rabelais Tours, Laboratoire d'Informatique, EA 2101

64 avenue Jean Portalis, 37200 Tours, France

nicolas.ragot@univ-tours.fr

---

*RÉSUMÉ.* Nous présentons dans cet article un nouvel algorithme pour l'adaptation des modèles de Markov cachés (modèles HMM). Le principe de notre algorithme d'adaptation itératif est d'alterner une phase d'adaptation de la structure des modèles HMM avec une phase d'adaptation des moyennes des gaussiennes des HMMs utilisant la méthode d'adaptation MAP. Cet algorithme est appliqué à la reconnaissance de caractères imprimés pour adapter les modèles de caractères appris par un moteur de reconnaissance polyfonte à de nouvelles formes de caractères. La comparaison des résultats obtenus avec ceux des adaptations classiques MAP et MLLR montre une légère augmentation des performances du système de reconnaissance.

*ABSTRACT.* We present in this paper a new algorithm for the adaptation of hidden Markov models (HMM models). The principle of our iterative adaptive algorithm is to alternate an HMM structure adaptation stage with an HMM Gaussian MAP adaptation stage. This algorithm is applied to the recognition of printed characters to adapt the models learned by a polyfont character recognition engine to new forms of characters. Comparing the results with those of MAP and MLLR classic adaptations shows a slight increase in the performance of the recognition system.

*MOTS-CLÉS :* Modèles de Markov cachés, adaptation de structure, reconnaissance de caractères, OCR.

*KEYWORDS:* Hidden Markov models, structure adaptation, character recognition, OCR.

---

## 1. Introduction

Bien qu'anciens, les modèles de Markov cachés (HMM [Rabiner 89]) sont toujours d'actualité et largement utilisés pour la modélisation de séquences probabilistes dans un vaste spectre d'applications, et tout particulièrement pour les applications de reconnaissance des formes (OCR, reconnaissance de la parole...). L'intérêt de cette technique réside dans le fait que l'approche Markovienne permet d'utiliser toute l'information disponible lors de la prise de décision en combinant un modèle d'attache aux données et un modèle des solutions recherchées (modèle de langage au sens large).

Un des principaux problèmes des méthodes d'apprentissage statistique (dont les HMM), est la dissimilarité entre les données d'apprentissage et les données de test. Les moteurs de reconnaissance de formes sont généralement construits de manière à reconnaître correctement la plus large variété de formes possible, ce qui donne par exemple, dans le cas de la reconnaissance de parole, des systèmes indépendants du locuteur, ou pour l'OCR des systèmes *polyfontes*. Il est également reconnu que ces systèmes génériques sont moins performants que des systèmes spécialisés sur un type particulier de forme comme par exemple un système *monofonte* dans le cas de l'OCR ou encore un système dépendant du locuteur pour la reconnaissance de la parole. Malheureusement, les données étiquetées nécessaires à l'apprentissage de tels systèmes font souvent défaut ou ne sont pas disponibles en quantité suffisante. Il est alors plus avantageux de modifier le système générique existant de manière à mieux reconnaître de nouvelles données. Cette opération de modification est appelée *adaptation*.

La quasi-totalité des algorithmes d'adaptation de modèles HMM modifient exclusivement la distribution de la probabilité d'émission (les Gaussiennes dans le cas de HMM à densité continue - CDHMM). La structure des HMM (le nombre d'états des modèles) reste quant à elle inchangée. Nous présentons dans cet article un nouvel algorithme qui adapte conjointement et de manière itérative la structure et les Gaussiennes des HMM à densité de probabilité continue. La méthode est détaillée et ses résultats comparés à ceux d'une adaptation classique des HMM selon les méthodes MAP (Maximum A Posteriori) [Gauvain et Lee 94] et MLLR (Maximum Likelihood Linear Regression) [Legetter et Woodland 95].

L'article est organisé comme suit. Dans la section 2, nous donnons un aperçu des méthodes d'apprentissage et d'adaptation des HMM. Dans la section 3, les méthodes d'adaptation de Gaussiennes MAP et MLLR sont détaillées. La section 4 contient une description de notre algorithme d'adaptation. Puis, dans la section 5, nous présentons le système de reconnaissance de caractères et les données utilisées lors de nos expérimentations. Les résultats sont détaillés dans la section 6.

## 2. Apprentissage et adaptation des modèles HMM

Modéliser une forme avec un HMM consiste en l'apprentissage statistique (à partir d'une base d'exemples) de deux paramètres : la distribution des données dans l'espace des caractéristiques (fonction de densité de probabilité d'émission : moyennes et matrices de covariances des gaussiennes dans le cas des CDHMM) et les probabilités de transition; un hyper-paramètre correspondant au nombre d'états des modèles. Il existe des algorithmes itératifs pour réaliser cette opération d'apprentissage, par exemple : Baum-Welch - basé sur l'algorithme EM – qui itère des phases d'alignement/réestimation des modèles sur les données d'apprentissage et Viterbi – qui cherche itérativement à maximiser la probabilité de génération des séquences d'apprentissage suivant leur chemin de Viterbi. Ces algorithmes optimisent les DDP (densités de probabilités) d'émission des modèles ainsi que les probabilités de transition mais la structure des modèles (nombre d'états) doit être déterminée préalablement par le concepteur. Des méthodes d'optimisation de la structure des modèles HMM ont déjà été proposées dans la littérature. Par exemple [Zimmermann et Bunke 02] proposent de déterminer la largeur du modèle en fonction du nombre moyen (sur la base d'apprentissage) de vecteurs de caractéristiques représentant une occurrence de la forme modélisée. La fraction  $X$  permettant de calculer le nombre d'états de chaque modèle à partir de la largeur moyenne de la forme modélisée est déterminée avec une approche de validation (la meilleure fraction est celle donnant le meilleur taux de reconnaissance).

Par ailleurs, certains auteurs proposent des algorithmes de *sélection de modèles* pour l'apprentissage des HMM. Ces algorithmes convergent vers une structure optimale à partir d'une structure initiale par une suite de fusions d'états [Stolcke et Omohundro 92] [Binsztok et Artières 05], de divisions d'états [Siddiqi et al. 07] ou par une succession de fusions et de divisions [Brants 96]. Dans ce dernier article, le principe de l'optimisation de la structure est de trouver à chaque itération l'opération (parmi l'ensemble des fusions possibles de deux états successifs ou des divisions d'un des états du modèle) produisant la plus grande augmentation du taux de reconnaissance sur une base de validation. L'algorithme est arrêté si aucune opération n'apporte d'amélioration du taux de reconnaissance par rapport au modèle actuel.

Les algorithmes d'adaptation développés jusqu'ici (comme par exemple en reconnaissance de la parole pour l'adaptation au locuteur et au bruit) ont uniquement eu pour but l'adaptation des densités de probabilité d'émission, généralement représentées par les Gaussiennes des modèles HMM. Les méthodes d'adaptation MAP et MLLR ont ainsi été développées dans ce but. Ces méthodes ont été par la suite modifiées, affinées et améliorées dans un grand nombre de publications parmi lesquelles on citera : [Wang et Zhao 01], [Povey et al. 03], [Bocchieri et al. 99]. Ces deux méthodes ont été appliquées avec succès à l'adaptation au scripteur d'un moteur de reconnaissance de l'écriture manuscrite [Vinciarelli et al. 04].

K. Ait-Mohand, L. Heutte, T. Paquet, N. Ragot

A notre connaissance, seul un auteur s'est intéressé à faire évoluer la structure des HMM lors d'un processus d'adaptation [Schambach 03]. Dans cette étude, l'auteur cherche à adapter des modèles HMM indépendants du scripteur à un scripteur particulier en trouvant la meilleure structure possible par une suite d'incrémentations d'états suivies d'un réapprentissage des modèles selon l'algorithme EM. Deux critères permettant de trouver la meilleure structure sont étudiés : un critère de vraisemblance normalisée et un critère de taux de reconnaissance (ce dernier donnant les meilleurs résultats). Ces critères rapprochent cette méthode des méthodes de sélection de modèles à l'apprentissage évoquées plus haut.

Pour développer notre algorithme d'adaptation, nous nous sommes basés sur une approche différente ; nous combinons des phases d'adaptation des lois d'émission des modèles selon les méthodes d'adaptation des Gaussiennes (MAP) avec des phases de modification de la structure où le nombre d'états de chaque modèle est estimé selon la largeur de la forme qu'il représente (qui est calculée d'après le nombre de trames sur lesquelles s'aligne chaque modèle, ce nombre étant moyenné sur toutes les données d'adaptation).

### 3. Adaptation des probabilités d'émission : méthodes MAP et MLLR

Nous nous limiterons au cas de modèles HMM de type gauche-droite à densité de probabilité d'émission continue représentées par des mélanges de Gaussiennes et nous supposons que les principales différences entre les ensembles de données sont caractérisées par les moyennes des Gaussiennes et nous ne modifierons en conséquence que ces moyennes. Les méthodes d'adaptation les plus utilisées dans ce cadre sont les méthodes MAP et MLLR.

L'adaptation MLLR consiste à trouver une matrice de transformation linéaire permettant de convertir le vecteur moyenne  $\mu_{im}$  de la Gaussienne  $m$  de l'état  $i$  d'un HMM en un vecteur moyenne adapté  $\mu'_{im}$ . La transformation réalise une combinaison linéaire des Gaussiennes du modèle initial pour calculer les nouvelles Gaussiennes. La transformation est calculée de manière à maximiser la vraisemblance des données d'adaptation par rapport au modèle selon un schéma EM. Pour augmenter la robustesse de la méthode, les Gaussiennes sont regroupées en *classes de régression* selon un critère de distance dans l'espace des caractéristiques. Les Gaussiennes d'une même classe de régression subissent la même transformation.

Le principe de l'adaptation MAP consiste, quant à lui, à converger de manière itérative vers un ensemble de paramètres optimal (les moyennes des Gaussiennes dans le cas présent) en réalisant une combinaison linéaire des paramètres précédents (initialisé avec les paramètres du modèle avant adaptation) avec ceux estimés sur les données d'adaptation selon le critère du maximum de vraisemblance selon la formule suivante :

$$\hat{\mu}_{im} = \frac{\sum_{t=1}^T \gamma_{im}(t) o_t}{\tau + \sum_{t=1}^T \gamma_{im}(t)} \bar{\mu}_{im} + \frac{\tau}{\tau + \sum_{t=1}^T \gamma_{im}(t)} \mu_{im} \quad [1]$$

$\hat{\mu}_{im}$  est la moyenne mise à jour.

$\mu_{im}$  est la moyenne à l'itération précédente.

$\bar{\mu}_{im}$  est la moyenne estimée sur les données d'adaptation.

$o_t$  est le vecteur au temps  $t$  d'une séquence issu d'une séquence d'observations de longueur  $T$  des données d'adaptation.

$\gamma_{im}(t)$  est la probabilité de la Gaussienne  $m$  de l'état  $i$  au temps  $t$ .

$\tau$  est une constante qui contrôle le poids relatif du modèle initial par rapport aux données d'adaptation.

Avec l'adaptation MAP, contrairement à MLLR, chaque moyenne est mise à jour séparément. Ce fait provoque une différence de comportement entre les deux méthodes : MAP exige une plus grande quantité de données d'adaptation que MLLR pour être efficace, mais si la quantité de données d'adaptation utilisée est suffisante, l'algorithme MAP est plus performant que MLLR.

#### 4. Algorithme d'adaptation conjointe de la structure et des Gaussiennes des modèles HMM

Nous allons à présent exposer notre algorithme combinant les méthodes d'adaptation des Gaussiennes exposées dans la section précédente avec une modification de la structure des modèles. L'algorithme alterne de manière itérative une phase d'adaptation des gaussiennes des HMM (selon l'une des méthodes MAP ou MLLR) avec une étape où la structure de chaque modèle est modifiée par ajout ou suppression d'un seul état.

##### 4.1. Modification de la structure des modèles

Supposons que la structure des modèles HMM ait été déterminée avant l'apprentissage en utilisant la technique développée dans [Zimmermann et Bunke 02]. L'étape de modification de la structure agit de la manière suivante :

- aligner les modèles existants sur la base de test ;
- calculer la largeur moyenne de chaque forme (modélisée par un des HMM) en terme de nombre de vecteurs de caractéristiques ;
- calculer la largeur de chaque modèle (en terme de nombre d'états) en multipliant la largeur moyenne de la forme modélisée par la fraction calculée lors de l'apprentissage ;

– si le nombre d'états trouvé est supérieur au nombre d'états actuel, ajouter un état au modèle. S'il est inférieur on diminue d'un état le nombre d'états du modèle. S'il est identique, la structure du HMM reste inchangée.

Pour ajouter un état supplémentaire à un HMM, on duplique l'état ayant la loi d'émission de plus grande variance car la plus grande variabilité de la forme représentée par cet état la rend plus susceptible d'être représentée plus finement par deux mélanges de Gaussiennes (et donc deux états) plutôt que par un seul mélange de Gaussiennes. On aura ainsi deux états successifs identiques. Les probabilités d'auto-transition des deux états seront modifiées de manière à ce que la longueur de chacun des deux états obtenus (la longueur d'un état étant définie comme étant égale à  $A/(1-A)$ ,  $A$  étant la probabilité d'auto-transition de l'état) soit égale à la moitié de la longueur de l'état dupliqué. Il est à noter que cette duplication des états rend inadéquate l'utilisation de l'adaptation MLLR. En effet, les mélanges de Gaussiennes des deux états créés par duplication étant identiques, ils seront regroupés dans une même classe de régression. Ils seront donc transformés de la même manière et resteront donc identiques après l'adaptation.

Pour supprimer un état à un HMM, on fusionne les deux états successifs dont les distributions de probabilité d'émission sont les plus proches (selon la distance de Kullback-Leibler). Les états ayant tous le même nombre de Gaussiennes, la loi d'émission de l'état fusionné est obtenue à partir de l'ensemble des Gaussiennes des deux états initiaux de manière itérative par fusion des deux Gaussiennes les plus proches jusqu'à obtention du nombre désiré de Gaussiennes. La probabilité d'auto-transition du nouvel état est calculée de manière à ce que la longueur de l'état soit égale à la somme des longueurs des deux états fusionnés.

#### **4.2. L'algorithme d'adaptation**

L'algorithme d'adaptation combine les deux étapes détaillées précédemment (adaptation des Gaussiennes et modification de la structure) de la manière suivante :

- 1– adapter les Gaussiennes des modèles initiaux selon la méthode MAP ;
- 2– modifier la structure des modèles ;
- 3– adapter les Gaussiennes ;

4– si la somme sur tous les modèles des différences entre le nombre d'états calculé et le nombre d'états actuel est inférieure à la moitié du nombre de modèles existant, arrêter l'itération, sinon, retour à l'étape 2.

Les deux états créés par le processus d'ajout d'un état vont se spécialiser après l'adaptation sur la portion de la forme (modélisée précédemment par un seul état) sur laquelle ils sont alignés. L'état produit par une opération de fusion modélise déjà (grâce à sa DDP d'émission qui est la fusion des 2 DDP de deux états) l'ensemble des deux formes précédemment représentées par deux états différents. L'adaptation des Gaussiennes permet ensuite d'affiner cette DDP de manière à mieux correspondre à la forme à modéliser.

## **5. Application à la reconnaissance de caractères imprimés**

Afin de valider notre algorithme d'adaptation, nous l'appliquons à un cas de reconnaissance de caractères imprimés. Nous avons créé un système de reconnaissance en utilisant des HMM pour modéliser les caractères.

### **5.1. Description du système de reconnaissance de caractères**

En entrée du système se trouve un extracteur de caractéristiques qui permet de convertir l'information contenue dans les images binarisées des lignes de texte en une séquence de vecteurs de caractéristiques. Cette opération est réalisée de la manière suivante : une fenêtre rectangulaire (de même hauteur que l'image de la ligne et de largeur optimisée par une approche de validation) balaye l'image de gauche à droite ; à chaque position (décalée de 2 pixels par rapport à la position précédente), on calcule sur cette fenêtre plusieurs caractéristiques (densités de pixels noirs, transitions entre pixels noirs et blancs, présence ou non de certaines formes élémentaires dans l'image) formant un vecteur de longueur fixe (voir [El-Hajj et al. 05]). La séquence de vecteurs de caractéristiques ainsi obtenue servira à représenter de manière compacte la séquence des formes contenues dans l'image. L'apprentissage des modèles se fait en appliquant l'algorithme de Baum-Welch sur les séquences de vecteurs de caractéristiques extraites d'un ensemble d'images d'apprentissage. Le nombre d'état de chaque modèle est déterminé en utilisant une base de validation selon la méthode détaillée dans [Zimmermann et Bunke 02]. Les autres paramètres de l'apprentissage (nombre de Gaussiennes par état des HMM et nombre d'itérations d'apprentissage) sont ensuite optimisés conjointement, toujours par une approche de validation.

Pour tester les performances du système, l'algorithme de Viterbi est appliqué à la séquence de vecteurs de caractéristiques extraites à partir d'une image de la base de test. L'algorithme de Viterbi calcule la séquence de caractères qui correspond le plus probablement à la séquence de vecteurs de caractéristiques fournie en entrée. Cette séquence de caractères est ensuite comparée à la séquence réelle (vérité-terrain, transcription par un opérateur humain du texte contenu dans l'image du document) afin de fournir un score (taux de reconnaissance, précision...) permettant de quantifier les performances du système.

### **5.2. Données utilisées**

Pour l'apprentissage et pour le test de notre système, nous avons créé notre propre base d'images à partir d'un fichier texte : chaque ligne de texte est convertie en une image de texte dans une police donnée et une taille de police fixée. Ces images sont ensuite dégradées (par la déformation des caractères et l'introduction de bruit) en utilisant le modèle de dégradation de Baird [Baird 90]. L'avantage d'utiliser des données artificielles étant de pouvoir directement disposer d'une base d'images de grande dimension et d'une grande variabilité de formes, avec la vérité-terrain associée, sans passer par une fastidieuse transcription manuelle.

K. Ait-Mohand, L. Heutte, T. Paquet, N. Ragot

Cette base contient des images à une résolution de 300 dpi dans 100 polices différentes, de la même taille de police (12 points). 70 de ces polices serviront à l'apprentissage du système et les 30 restantes à tester l'aptitude du système à la généralisation. Les polices ont été choisies de manière à couvrir également toutes les familles de fontes existantes. Selon la classification des polices établie par l'association typographique internationale (classification Vox-Atypi [Laliberté 04]) il y a dix familles de polices différentes (Humanes, Garaldes, Réales, Didones, Mécanes, Linéales, Incises, Scriptes, Manuaires, Gothiques – voir tableau 1). La figure 1 ci-dessous montre quelques exemples d'images tirées de cette base :

**fier, levant dans la nuit son cimier flamboyant,**  
pour tout bruit, le frisson lugubre que fait l'onde  
de lechus, de platon, d'othon, d'ursus, d'étienne,  
ouvrait les deux battants de sa porte sonore .  
**voyaient jadis, parmi leurs ombres vénérables,**

Figure1. Exemples d'images synthétiques

Famille de polices	Polices de caractères		
Humanes	Berkeley old style	Legacy serif	Deepdene
Garaldes	Fournier	Times	Georgia
Réales	Aldus	Caslon	Perpetua
Didones	Modern 216	URW antiqua	Bauer Bodoni
Mécanes	Egyptienne F.	Glypha	Melior
Linéales	<b>Folio</b>	Akzidenz grotesk	Frutiger
Incises	Friz Quadrata	Incised 901	Poppl laudatio
Scriptes	<i>Mistral</i>	<i>Park avenue</i>	<i>Present</i>
Manuaires	<b>Banco</b>	<b>Ondine</b>	<b>Bolide</b>
Gothiques	<b>Sebalduš_Gotisch</b>	<b>Fette Becker gotisch</b>	<b>Fette Kanzlei</b>

Tableau 1. Polices de caractères de la base de test.

## 6. Expérimentations et résultats

Une fois l'apprentissage du système de reconnaissance polyfonte réalisé sur la base d'apprentissage contenant des images de lignes de 70 polices différentes, nous testons notre système sur la base de test qui contient des images de 30 polices de caractères différentes de celles de la base d'apprentissage. Le système polyfonte est ensuite adapté successivement à chacune des 30 polices de test en utilisant les deux méthodes différentes : une adaptation des Gaussiennes exclusivement selon la méthode MAP utilisée dans les modes supervisé (les images de la base d'adaptation sont accompagnées de leur transcription) et non-supervisé (les transcriptions sont créées en utilisant les modèles existants) ; notre algorithme d'adaptation conjointe des Gaussiennes et de la structure des HMM. Les bases d'adaptation utilisées consistent chacune en une dizaine d'images de lignes (soit un total de 500 caractères en moyenne) de la police de caractères à laquelle on souhaite adapter le système. Chacune de ces bases d'adaptation est également utilisée comme base d'apprentissage d'un système monofonte (dédié à une seule fonte) afin de comparer ses performances à celles du système polyfonte adapté à cette même fonte.

Afin de mesurer l'apport du changement de la structure des modèles durant l'exécution de l'algorithme, nous avons aussi testé une méthode d'adaptation MAP itérative dans laquelle le modèle adapté une première fois sert ensuite de modèle initial à une nouvelle adaptation MAP et ainsi de suite jusqu'à un certain nombre d'itérations (ici 4 itérations, ce qui correspond au nombre moyen d'itérations de notre algorithme itératif observé sur la base de test). Les résultats comparés des 3 méthodes sont donnés ci-dessous. Le tableau 2 donne le taux de reconnaissance de caractères moyen sur toutes les polices de test tandis que le tableau 3 détaille pour l'adaptation supervisée les résultats les plus significatifs pour quelques polices de caractères.

type d'adaptation	sans adaptation	adaptation MAP non-supervisée	adaptation MAP supervisée
mélanges de Gaussiennes uniquement	83.83%	85.62%	93.49%
mélanges de Gaussiennes uniquement (4 itérations)		85.69%	95.57%
mélanges de Gaussiennes et structure		85.22%	95.02%

**Tableau 2.** Résultats du système avant et après adaptation (pour les 3 types d'adaptation différents). Les résultats sont moyennés sur les 30 polices de caractères différentes et sont données en pourcentage du taux de reconnaissance caractère.

police de caractères	sans adaptation	mélanges de Gaussiennes uniquement	mélanges de Gaussiennes uniquement (4 itérations)	mélanges de Gaussiennes et structure
Berkeley old style	96.62%	97.66%	97.46%	94.51%
<b>Banco</b>	34.08%	57.14%	67.47%	71.24%
<i>Mistral</i>	46.63%	80.61%	92.59%	91.78%
<b>Fette Sanserif</b>	68.36%	90%	93.94%	95.74%

**Tableau 3.** Résultats du système avant et après adaptation (en mode supervisé) pour 4 polices de caractères différentes.

Les trois méthodes d'adaptation améliorent nettement les résultats de reconnaissance du système polyfonte. Elles sont aussi bien meilleures que les systèmes monofontes directement appris sur les seules données ayant servi à l'adaptation du polyfonte (qui donnent un taux de reconnaissance de 83.5% en moyenne). Les résultats obtenus par notre algorithme, moyennés sur toutes les polices, sont du même ordre que ceux obtenus avec plusieurs itérations d'adaptation MAP. L'analyse des résultats détaillés par polices dans le tableau 2 montre cependant des différences de comportement notables sur certaines des polices de caractères testées.

Le fait de répéter plusieurs fois l'adaptation des Gaussiennes avec la méthode MAP améliore nettement dans la plupart des cas les résultats de l'adaptation. Notre algorithme d'adaptation de structure est lui aussi plus performant que l'algorithme MAP ordinaire mais ses performances restent le plus souvent inférieures à celles du MAP itératif même si elles sont parfois meilleures sur certaines polices, notamment les polices de caractères très mal reconnues par le système de reconnaissance initial. Une explication possible à ce comportement est la « brutalité » du changement de structure observée sur certaines de ces polices, les modèles de caractères obtenus comptant parfois deux fois plus d'états que les modèles initiaux. L'adaptation des Gaussiennes n'arrive plus à compenser cette importante modification de la structure du modèle initial.

En résumé, notre algorithme montre un comportement proche de celui du système adapté avec plusieurs itérations de MAP et l'apport du changement de la structure des modèles HMM n'est intéressant que sur certaines polices.

## 7. Conclusion

Nous avons proposé un algorithme permettant, tout en adaptant la loi d'émission de modèles HMM, d'adapter en même temps leur structure de manière à

mieux tenir compte des spécificités des données sur lesquelles on souhaite s'adapter. L'application de cet algorithme à la reconnaissance de caractères donne en moyenne des résultats similaires à plusieurs adaptations successives des lois d'émission des modèles. Alors qu'il peut permettre d'améliorer très nettement les résultats sur les polices très mal reconnues, il peut aussi dégrader un peu les performances par rapport à une adaptation classique sur les polices pour lesquelles l'adaptation des lois d'émission fonctionne déjà bien.

Cela nous permet d'envisager plusieurs pistes d'améliorations liées aux points faibles de la méthode que nous avons identifiés. Le premier concerne le critère que nous avons utilisé pour guider cette adaptation de la structure : la largeur des caractères mesurée par les modèles (en utilisant une fraction, calculée sur des données différentes de celles auxquelles on souhaite adapter nos modèles). Celui-ci n'est sans doute pas pertinent car optimisé sur les données d'apprentissage et peu adaptable. Il serait possible de remédier à ce défaut en utilisant par exemple des HMM à modèle de durée qui eux pourraient être adaptés aux données à traiter. Le deuxième défaut provient du fait d'itérer systématiquement l'algorithme MAP car cela conduit à sur-apprendre certains modèles initialement assez proches des statistiques des données d'adaptation. Il serait judicieux dans ce cas de proportionner l'intensité de l'adaptation à la distance entre les modèles à adapter et les statistiques des données d'adaptation.

## 12. Bibliographie

- [Baird 90] Baird H. S., Document Image Defect Models, *IAPR Workshop on SSPR*, Murray Hill, Etats-Unis, 13-15 Juin 1990.
- [Binsztok et Artières 05] Binsztok H., Artières, T., Learning Model Structure from Data: An Application to On-Line Handwriting, *ELCVIA*, 5(2), p. 30-46, 2005.
- [Bocchieri et al. 99] Bocchieri E., Digalakis V., Corduneanu A., Boulis C., Correlation modeling of MLLR transform biases for rapid HMM adaptation to new speakers, *Proc. ICASSP*, vol. 2, p. 773-776, 1999.
- [Brants 96] Brants, T., Estimating Markov Model Structures, *Proceedings of the Fourth Conference on Spoken Language Processing (ICSLP-96)*. Philadelphie, Etats-Unis, p. 893-896, 1996.
- [El-Hajj et al. 05] El-Hajj R., Likforman-Sulem L., Mokbel C., Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling, *International Conference on Document Analysis and Recognition (ICDAR), 2005*, p. 893-897.
- [Gauvain et Lee 94] Gauvain J-L., Lee C.H., Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, *IEEE Transactions on Speech and Audio Processing* 2, p. 291-298, 1994.
- [Laliberté 04] Laliberté J., *Formes typographiques: historique, anatomie, classification*, Presses de l'Université Laval, 2004.

K. Ait-Mohand, L. Heutte, T. Paquet, N. Ragot

- [Legetter et Woodland 94] Legetter C. J., Woodland P. C., Maximum Likelihood linear regression for speaker adaptation of continuous density HMM's, *Computer, Speech and Language*, vol. 9, p. 171-186, 1995.
- [Povey et al. 03] Povey D., Gales M.J.F., Kim D.Y., Woodland P.C., MMI-MAP and MPE-MAP for acoustic model adaptation, Proc. EUROSPEECH, Genève, p. 1981–1984, 2003.
- [Rabiner 89] Rabiner L., A tutorial on hidden markov models and selected applications in speech recognition, *Proc. IEEE* 77(2), p. 257-285, 1989.
- [Schambach 03] Schambach, M., Model Length Adaptation of an HMM based Cursive Word Recognition System. *Proceedings of the Seventh international Conference on Document Analysis and Recognition (ICDAR) - Volume 1*, IEEE Computer Society, Washington, Etats-Unis, p. 109, 2003.
- [Siddiqi et al. 07] Siddiqi S., Gordon G., Moore A., Fast state discovery for HMM model selection and learning. *International Conference on Artificial Intelligence and Statistics 11*, p. 492–499, 2007.
- [Stolcke et Omohundro 92] Stolcke, A., Omohundro, S. M., Hidden Markov Model Induction by Bayesian Model Merging, *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds. Morgan Kaufmann, San Francisco, Etats-Unis, p. 11-18, 1992.
- [Vinciarelli et al. 04] Vinciarelli, A., Bengio, S., Bunke, H., Offline Recognition of Unconstrained Handwritten Texts Using HMMs and Statistical Language Models, *IEEE Trans. Pattern Anal. Mach. Intell.* 26(6), p. 709-720, 2004.
- [Wang et Zhao 01] Wang S. et Zhao Y., Online bayesian tree-structured transformation of HMMs with optimal model selection for speaker adaptation, *IEEE Trans. Speech Audio Processing*, vol. 9, p. 663–677, Sept. 2001.
- [Zimmermann et Bunke 02] Zimmermann M., Bunke H., Hidden Markov Model Length Optimization for Handwriting Recognition Systems, *International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, 2002, p. 369-374.