Partitionnement de tracés manuscrits en ligne par modèles markoviens

Henri Binsztok, Thierry Artières, Patrick Gallinari

Partitionnement de tracés manuscrits en ligne par modèles markoviens

Henri Binsztok – Thierry Artières – Patrick Gallinari

Laboratoire d'Informatique de Paris 6 (LIP6)
8, rue du Capitaine Scott
75015 Paris, France
email : prenom.nom@lip6.fr

Résumé : *Nous présentons une approche pour le partitionnement non supervisé de séquences. Cette méthode est inspirée de méthodes d'apprentissage de la topologie de modèles markoviens et repose sur la définition d'une distance entre modèles de Markov. Ce type de technique peut être utilisé pour apprendre, à partir des données, des modèles de caractères markoviens ou bien pour identifier des allographes ou des styles d'écriture en ligne.*

Abstract : *We present an unsupervised approach to cluster sequences. This method is inspired by topology learning methods for hidden Markov models, and is built upon the definition of a distance between Markov models. This type of technique may be used to learn Markovian character models from data or to identify allographs or handwriting styles.*

Mots-clés : Modèles de Markov cachés (MMC), Allographes, Ecriture en ligne, Partitionnement de séquences
Keywords : Hidden Markov Models (HMM), Allographs, Online handwriting, Sequence clustering

1 Introduction

Nous nous plaçons dans le cadre du développement de systèmes markoviens de reconnaissance de l'écriture manuscrite en ligne et explorons la possibilité d'apprendre la structure des modèles des caractères automatiquement à partir des données. L'apprentissage de modèles de Markov cachés (MMC) est généralement réalisé en deux étapes, un choix a priori d'une structure de modèle, puis un apprentissage statistique des paramètres à partir d'une base de données. Quelques approches ont été proposées dans le domaine de l'écrit pour automatiser, d'une façon limitée, le choix a priori des modèles, notamment sur le nombre d'états. Des méthodes plus génériques ont été proposées pour l'apprentissage de la structure de MMC mais leur généralité ne les rend pas nécessairement performantes pour le traitement des signaux écrits en ligne. Nous abordons le problème de l'apprentissage de structure comme un problème de partitionnement de données séquentielles en développant une méthode qui permet simultanément de partitionner des séquences d'apprentissage et d'apprendre des MMC gauche-droite pour les partitions. Notre approche est une approche non supervisée, guidée par les données. Elle permet l'apprentissage de la topologie de modèles de caractères et peut être utilisée en particulier pour identifier des allographes ou partitionner des scripteurs suivant leurs styles d'écriture. Cette dernière problématique n'est pas nouvelle. [PRE 00] propose une approche performante en quatre étapes : segmentation des caractères en *tracés* élémentaires, puis agglomération autour de prototypes - environ 1 exemple sur 5. Ensuite, l'agglomération est relancée sur les prototypes. L'approche est validée via un classifieur. Plus récemment, [NOS 03] choisit une approche probabiliste pour définir une partition de motifs. Pour chaque caractère, une approche semblable à EM est utilisée pour apprendre les probabilités qu'un caractère appartienne à une partition donnée. L'association du partitionnement et de modèle MMC a également été abordée par [PER 00] et [LOC 93]. Ce dernier propose de déterminer le nombre d'états et la structure du modèle par un algorithme itératif appliqué à la reconnaissance de la parole. Enfin, des approches de partitionnement hiérarchique appliquées au problème de la sélection d'allographes ont été étudiées dans [VUU 97].

Notre approche est une étude préliminaire que nous souhaitons étendre à l'avenir à l'apprentissage automatique de graphèmes dans des bases de signaux écrits. Pour cette raison, nous avons choisi de nous inspirer de travaux plus généraux sur l'apprentissage de structures de MMC, plus facilement extensibles à cette tâche. La stratégie adoptée consiste tout d'abord à construire un MMC initial à partir de toutes les données d'apprentissage, ce MMC étant composé d'autant de MMC gauche-droite (branches) qu'il y a de séquences d'apprentissage. Ce modèle est ensuite simplifié itérativement en fusionnant les branches par un algorithme similaire à un algorithme de partitionnement. Le critère employé lors de la fusion repose sur l'introduction d'une nouvelle mesure de similarité entre MMC gauche-droite.

Nous présentons tout d'abord la construction du modèle initial à partir des données (section 2). Puis, nous présentons notre algorithme de simplification itératif (section 3) en détaillant la distance entre MMC utilisée (section 4). Nous fournissons ensuite des résultats expérimentaux (section 5) visant à mettre en évidence la capacité de notre algorithme à identifier et modéliser des partitions dans une base de séquences. Même si notre approche peut être utilisée pour apprendre la toplogie d'un modèle de caractère markovien et du coup identifier ses allographes, nous avons choisi de réaliser nos expériences sur des bases de signaux extraites de la base Unipen [GUY 94], et contenant des tracés de chiffres divers et ressemblant (0 et 9 notamment). L'évaluation du partitionne-

ment est en effet beaucoup plus aisé dans ce cas puisque l'on dispose pour chaque exemple de l'information sur le chiffre tracé, alors que nous ne disposons pas de base de données d'allographes étiquettée.

2 Modélisation des données

Nous discutons ici de notre approche en commençant par la situer dans le cadre de l'apprentissage de la topologie de MMC. Nous présentons ensuite la procédure de construction du modèle MMC initial dont nous abordons la simplification en section 3.

2.1 Apprentissage de la structure

Les approches visant à déterminer la structure d'un MMC sont peu nombreuses et sont soit développées de façon *ad-hoc* pour une tâche particulière, soit trop générales pour etre performantes sur tout type de données. Parmi les approches proposées, une des plus intéressantes est décrite dans [STO 93]. A l'aide d'un algorithme "top-down", les auteurs proposent de construire un MMC initial complexe à partir des données. Ensuite, des états de ce MMC sont fusionnés un à un tant que la perte de vraisemblance des données par le modèle n'est pas trop importante. Une autre approche est proposée par [BRA 99] dans laquelle la simplification, itérative, du modèle est fondée sur des probabilités *a priori* entropiques des transitions entre états. Une des propriétés de cette approche est qu'une partie des probabilités de transition convergent vers 0. Certains états, devenant "injoignables", sont retirés de la structure du modèle. Nous nous proposons d'adapter la démarche de [STO 93] en restreignant la topologie à un mélange de modèles gauche-droite, chacun de ces modèles correspondant à une partition des séquences (e.g. un allographe). Notre approche consiste, comme celle de Stolcke, à construire un modèle probabiliste initial à partir des données, puis à le simplifier itérativement. Nous décrivons maintenant comment construire ce modèle initial.

2.2 Des données au modèle MMC initial

Cette première étape consiste à construire un modèle MMC résumant l'ensemble des séquences d'apprentissage. Nous avons choisi, plutôt que de travailler sur une représentation de bas niveau comme une séquence temporelle de points ou de trames, d'utiliser une représentation de "haut niveau" du signal proposée dans [ART 02]. Dans ce système, un signal d'écriture est transformé en une séquence de tracés élémentaires représentant au mieux le tracé originel. Un alphabet, Σ, de 36 tracés élémentaires, représentés figure 1, est utilisé pour cela, il comporte 12 tracés droits dans des directions uniformément réparties entre 0 et 360°, 12 courbes convexes et 12 courbes concaves. La séquence de tracés représentant au mieux un signal est déterminée par un algorithme de programmation dynamique dans un MMC ergodique, dont chacun des états modélise un des tracés élémentaires. Ainsi, le tracé d'un caractère 'e' peut être représenté par la séquence de tracés élémentaires $mnquvm$, en utilisant la dénomination des tracés de l'alphabet de la figure 1.

De la même façon que dans [STO 93] ou [MAR 03], un MMC gauche-droite est construit à partir de chaque tracé d'apprentissage. Par exemple, à partir du tracé du 'e' précédent, constitué de 6 tracés élémentaires, on construit un MMC gauche-droite à 6 états. Pour les lois de probabilité d'émission, définies sur l'alphabet Σ des tracés élémentaires, l'approche proposée par [LOC 93], consistant à apprendre les probabilités avec un algorithme standard pour les MMC ne nous a pas paru pertinente. En effet, cet apprentissage est délicat dans la mesure où il est indispensable de trouver une bonne initialisation. Nous avons choisi plutôt de partager, comme dans [MAR 03], les lois de probabilité d'émission dans des états correspondant au même tracé élémentaire. Par exemple le premier état et le dernier état du modèle construit à partir du 'e' précédent correspondent idéalement à un tracé élémentaire m, et partagent donc la même loi de probabilité. Il n'y a donc que 36 lois de probabilités à estimer (i.e. cardΣ). Nous avons envisagé deux stratégies pour définir ces lois de probabilités d'émission. La première, similaire à celle de [MAR 03], consiste à fixer par des connaissances a priori les lois de probabilités. Par exemple, il est possible de définir une similarité relativement intuitive entre tracés élémentaires, fonction de l'angle et de la courbure de ces tracés. La seconde stratégie consiste à apprendre ces lois de probabilités à partir des séquences d'apprentissage. Nous détaillons maintenant cette possibilité. L'idée est assez simple et consiste à considérer comme similaires des symboles (i.e. tracés de Σ) qui apparaissent dans le même type de séquences. Plus spécifiquement, on détermine pour chaque symbole de Σ sa fréquence d'apparition après tout préfixe (e.g. les préfixes de la représentation du 'e' précédent sont m, mn, mnq, nq, etc.), on obtient ainsi un vecteur de fréquences d'apparition du symbole pour chaque préfixe observé. Pour faciliter l'estimation, nous nous sommes limité à des préfixes de longueur 1. On définit la similarité entre deux symboles de Σ par la corrélation entre les deux vecteurs les représentant. On obtient alors les probabilités d'émission par normalisation. Nous verrons expérimentalement dans la section 5 que cette estimation fournit de bons résultats.

Le modèle MMC initial est un modèle MMC constitué de plusieurs branches, chaque branche correspondant au MMC gauche droite construit à partir d'une séquence d'apprentissage. Ce MMC possède un état initial et un état final dont on ne sort pas ; il existe des transitions de l'état initial vers les premiers états des modèles gauche-droite (les branches). Chaque état de ces modèles possède une transition soit vers lui-même, soit vers l'état suivant dans le modèle. Ce modèle MMC initial peut donc être vu comme un modèle de mélange des MMC gauche-droite construits à partir des séquences d'apprentissage. La particularité de notre approche consiste dans ce choix a priori de la topologie du MMC initial. En choisissant une topologie du type modèle de mélange de MMC gauche-droite, on force l'apprentissage à réellement identifier des types de séquences.

3 Simplification du MMC

La simplification du MMC initial décrit en section précédente est réalisée de façon non supervisée. Le schéma général de cet algorithme est de considérer l'ensemble des branches constituant le modèle de mélange et de fusionner itérativement les deux branches présentant la plus forte similarité. Nous présentons cette similarité, δ, entre MMC gauche -

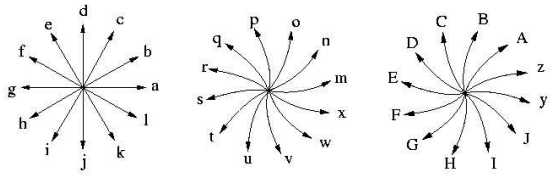

FIG. 1 – Alphabet des tracés élémentaires de Σ

droite en section suivante. Dans un premier temps, nous décrivons l'algorithme, puis nous détaillons le critère de sélection C dont dépend le critère d'arrêt, qui permet de déterminer automatiquement le nombre de branches. L'algorithme est le suivant :

1. Pour chaque séquence de la base, construire le MMC gauche-droite correspondant. On construit alors le modèle initial E.
2. Tant que le critère d'arrêt est optimisé :
   (a) Calculer les distances δ entre toutes les branches du modèle.
   (b) Sélectionner le couple de branches (u, v) les plus proches au sens de la distance δ.
   (c) Conserver parmi les deux nouveaux MMC candidats $E \setminus \{u\}$ et $E \setminus \{v\}$ le MMC qui optimise le critère C.
   (d) Retourner en 2.

L'approche proposée est une approche de sélection de modèles, puisque les composantes du MMC initial sont soit éliminées soit conservées par l'algorithme de simplification itérative. C'est un cas particulier de l'approche qui consiste à déterminer un modèle fusion des branches u et v. Si l'algorithme de construction du MMC initial puis de simplification itérative est appliqué à un ensemble de signaux peuvent correspondre à un caractère, les branches finales du modèle correspondent à des modèles des allographes.

On peut, pour chaque exemple de la base d'apprentissage, déterminer quelle branche a la plus forte probabilité d'avoir généré cet exemple. Ainsi, sur cette base, on peut grouper tous les exemples qui sont générés par le même modèle dans une même partition. C'est ce type de procédure que nous avons utilisée pour obtenir automatiquement les partitions présentées dans les figures 2 et 5.

Plusieurs critères d'évaluation sont possibles pour le MMC final obtenu. L'utilisation de modèles probabilistes fournit une option évidente : celle d'utiliser la vraisemblance des données avec la famille de modèles considérée. Il est possible de pénaliser ce critère par la taille du modèle, pour privilégier un modèle compact. Nous avons choisi un critère de ce type, la longueur de description minimale [RIS 82]. Ce critère s'exprime comme : $C = \log P(D|E) - \alpha \log[n|\Sigma| + h(D)]$ où $P(D|E)$ est la vraisemblance des données par le MMC, n le nombre total d'états du MMC, $|\Sigma|$ le cardinal de l'alphabet utilisé pour représenter les séquences, $h(D)$ la taille de description des données par le MMC, qui est constante dans notre cas. α est un paramètre permettant de régler l'importance des deux termes du critère, la finesse de modélisation et la complexité du modèle. Le critère d'arrêt est satisfait lorsque le critère C cesse de croître.

4 Mesure de similarité entre MMC gauche-droite

Soit M_1et M_2 deux MMC gauche-droite, de longueur respectives n et m. Nous proposons une mesure de similarité entre ces deux modèles prenant en compte la topologie gauche-droite de ces modèles. Cette mesure est fondée sur un alignement entre les états de M_1 et M_2. Puisque l'on cherche à déterminer une distance entre modèles, nous avons choisi de définir cette distance à partir d'une distance entre lois de probabilités d'émission associées aux états des deux modèles. Nous avons utilisé une distance classique pour cela, la distance de Kullback-Leibler symétrisée. Nous utilisons ces distances comme des coûts locaux dans un algorithme d'alignement temporel (Dynamic Time Warping en anglais). On cherche donc à trouver un alignement entre les états de M_1 et les états de M_2 qui minimise le coût :

$$J = \sum_{k=1}^{p} d_{KL}(i_k, j_k)$$

où $d_{KL}(i_k, j_k)$ est la distance de Kullback symétrisée entre les distributions de probabilités de l'état i_k de M_1 et l'état j_k de M_2 et où la séquence des indices $\{(i_k, j_k), k \in [1, p]\}$ correspond au chemin suivi pour apparier les états des deux modèles sous contraintes. Comme nous nous intéressons à des MMC gauche-droite, nous imposons les conditions limites $(i_1, j_1) = (1, 1)$, $(i_p, j_p) = (n, m)$ et pour tout k, $i_{k+1} \in \{i_k, i_k + 1\}$ et $j_{k+1} \in \{j_k, j_k + 1\}$. On définit alors la similarité entre séquences par :

$$\delta(M_1, M_2) = \hat{J} = \min_{p, \{(i_k, j_k), k \in [1,p]\}} \sum_{k=1}^{p} d_{KL}(i_k, j_k)$$

5 Résultats expérimentaux

Bien que notre approche puisse être utilisée pour identifier des allographes d'un même caractère, nous réalisons toute une série d'expériences sur des tracés de chiffres différents et ressemblants ('0' et '9' notamment). Ceci nous permet beaucoup plus facilement d'évaluer la pertinence des partitions obtenues, puisque l'on dispose pour chaque exemple de l'information sur le chiffre tracé, alors que nous ne disposons pas de base de données d'allographes étiquettée. Nous conclurons par une application à l'identification des allographes d'un même chiffre, sans évaluation numérique de la performance. Ces chiffres manuscrits sont extraits de la base Unipen [GUY 94]. Chaque signal d'écriture a été converti en une séquence de tracés élémentaires comme décrit en section 2.2. Nous n'avons pas pour le moment pris en compte l'information de levée de stylo, aussi nous n'avons utilisé que des caractères manuscrits écrits sans levée de stylo. Nous présentons maintenant les critères d'évaluation de nos algorithmes, puis nous fournissons des résultats expérimentaux, en comparant notre méthode à une méthode de référence pour le partitionnement de séquences.

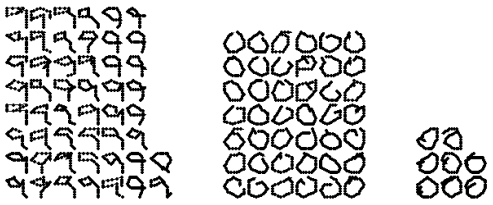

FIG. 2 – Représentation des modèles de séquence d'écriture manuscrite après traitement. Ces exemples des chiffres '0' et '9' ont été automatiquement partitionnés par notre approche avec $\alpha = 2.5$.

5.1 Méthodes d'évaluation

Nous utilisons deux méthodes pour évaluer la performance de notre partitionnement. La première, classique pour les algorithmes de partitionnement, consiste à calculer deux estimateurs : une mesure entropique et une mesure appelée mesure F. La seconde est visuelle et nous montrons le résultat d'une expérience de partitionnement obtenu pour des tracés des chiffres 0 et 9.

Le résultat de l'algorithme de classement est un ensemble de séquences groupées dans des partitions. Comme le rappelle [STE 00], il existe deux mesures complémentaires permettant d'évaluer la pertinence d'un partitionnement. Ces mesures d'évaluation sont exploitables pourvu que l'on dispose d'une information sur les classes réelles des formes traitées, c'est la raison pour laquelle nous ferons des expériences sur des données mélangeant des tracés de plusieurs chiffres. Dans la suite, nous nommerons partitions le résultat du partitionnement et classes l'étiquetage des données. La première mesure, nommée "entropie totale", est liée à l'homogénéité des partitions par rapport à l'information de classe. Si toutes les séquences d'une partition correspondent à une même classe, les partitions sont parfaitement homogènes, et l'entropie est nulle et minimale. Plus formellement, pour une partition j, l'entropie est définie par

$$E_j = -\sum_i p_{ij} \log p_{ij},$$

où p_{ij} est la probabilité qu'un élément de la partition j appartienne à la classe i, elle est estimée par des comptages. L'entropie totale est fournie par une somme pondérée sur les différentes partitions :

$$E_T = \sum_{j=1}^{n} \frac{n_j E_j}{n},$$

où n_j est le nombre d'éléments de la partition j et n le nombre total d'exemples. La seconde mesure est appelée mesure F, elle est classique dans le domaine de la recherche d'informations. Si toutes les partitions sont homogènes et non redondantes (il n'existe pas deux partitions correspondant à une même classe), la mesure F vaut 1 et est maximale. La mesure F agrège deux mesures : la précision et le rappel. La précision capture une information analogue à l'entropie, tandis que le rappel est élevé si les partitions sont homogènes mais trop éparpillées. Par définition, $P(i,j) = \frac{n_{ij}}{n_j}$ et $R(i,j) = \frac{n_{ij}}{n_i}$. Alors, $F(i,j) = \frac{2P(i,j)R(i,j)}{P(i,j)+R(i,j)}$. La mesure F est calculée pour chacune des classes puis on calcule une moyenne pondérée sur les classes : $F = \sum_i \frac{n_i}{n} \max\{F(i,j)\}$.

5.2 Résultats comparés

Nous présentons ici divers résultats expérimentaux. Les premiers concernent l'estimation des lois de probabilités d'émission comme décrit en section 2.2. Nous comparons ensuite les résultats de notre algorithme de partitionnement à une approche récente de partitionnement de séquences. Comme nous l'avons déjà mentionné, nos expériences de partitionnement sont réalisées en non supervisé sur des ensembles d'exemples de plusieurs chiffres (e.g. 0 et 9) afin de pouvoir être évaluées quantitativement.

5.2.1 Estimation des lois de probabilités dans le MMC initial

La figure 3 représente les valeurs de la matrice de similarité entre symboles de Σ determinée manuellement suivant des connaissances a priori, ainsi que les valeurs de la matrice de similarité estimée par la méthode présentée en section 2.2. Ces matrices sont des matrices 36x36 : en ordonnée et en abscisse figurent les 36 modèles de tracés ; le niveau de gris est proportionnel à la similarité (blanc = forte similarité, noir = faible similarité). Nous constatons une forte ressemblance

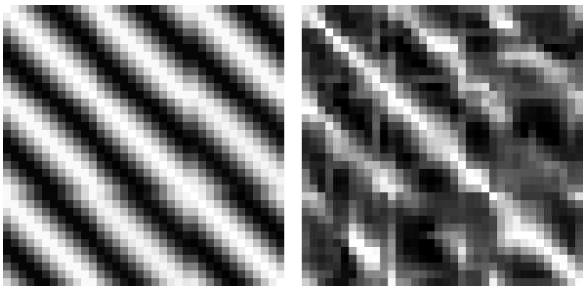

FIG. 3 – Matrices de similarités entre symboles de Σ, fixée (à gauche) et apprise (à droite)

entre ces matrices, ce qui montre que la méthode d'estimation permet de capturer raisonnablement l'information de similarité entre symboles contenue dans les séquences de la base.

5.2.2 Expériences en partitionnement

Nous avons comparé notre approche à une approche récente de classement de séquences proposée par [CAD 00] et fondée sur l'algorithme EM. L'algorithme de Cadez étant fortement dépendant de l'initialisation (une initialisation aléatoire fournissant de mauvais résultats), nous avons initialisé cette méthode avec le résultat de notre approche. Nous avons réalisé plusieurs expériences. Dans une première expérience, (EXP1), nous utilisons 100 exemples des chiffres manuscrits '0' et '9', chiffres qui présentent une forte similarité. Dans les deux séries d'expériences suivantes (EXP2 et EXP3) nous utilisons 150 exemples de tracés des chiffres '0', '1' et '2'. Le partitionnement étant non supervisé, les exemples sont

	Entropie : BAG	Entropie : Cadez
EXP1 *moyenne*	0.14	0.27
écart-type	0.12	0.13
EXP2 *moyenne*	0.32	0.20
écart-type	0.16	0.03
EXP3 *moyenne*	0.13	0.18
écart-type	0.13	0.06
	F : BAG	F : Cadez
EXP1 *moyenne*	0.82	0.74
écart-type	0.13	0.16
EXP2 *moyenne*	0.63	0.58
écart-type	0.12	0.09
EXP3 *moyenne*	0.62	0.52
écart-type	0.04	0.06

TAB. 1 – Performances comparées (entropie et mesure F) de notre approche (BAG) et [CAD 00] (Cadez).

partitionnés sans séparer apprentissage et test comme c'est le cas pour les problèmes supervisés. Les expériences EXP2 et EXP3 diffèrent par la valeur de l'hyper-paramètre α (1.5 pour EXP2, et 1 pour EXP3). Pour chacune de ces trois séries d'expériences, nous avons réalisé différents essais et moyenné les résultats, en collectant les écart-types sur ces performances. Les résultats de ces expériences sont fournis dans le tableau 1. Dans tous les cas, on note une amélioration significative des performances de notre approche par rapport à celle de [CAD 00], du point de vue de l'homogénéité des partitions (faible entropie) et du nombre limité de partitions trouvées (mesure F proche de 1). En comparant les résultats des deux dernières séries d'expériences, différant par la valeur de α, on voit que les résultats sont assez similaires du point de vue de la mesure F, mais l'entropie est plus élevée pour l'expérience EXP2, utilisant une valeur plus forte. Cela est naturel puisque le terme de pénalisation sur la complexité du système est plus forte pour EXP2, résultant en un plus faible nombre de partitions.

Une autre série d'expériences a été effectuée en utilisant une représentation incluant une information de durée. Cette information est prise en compte en multipliant les symboles dans la représentation en fonction de la longueur des tracés auxquels ils correspondent. Par exemple, pour un tracé représenté par la séquence ivm, le troisième tracé étant deux fois plus long que les deux premiers, la représentation utilisée ici serait $ivmm$. Ces résultats sont présentés dans la figure 4 et ont été effectués sur les mêmes chiffres que ceux de l'expérience EXP1. On note une amélioration très significative des performances par rapport aux séquences sans information de durée. Notamment, pour $\alpha = 2.5$, l'entropie est nulle et la mesure F supérieure à 80%.

C'est le partionnement issu de cette expérience qui est présenté dans la figure 2. On voit bien sur cet exemple, que les partitions trouvées sont homogènes (que des 0 ou que des 9), même si deux partitions sont trouvées pour le chiffre 0, ce qui s'explique par la variabilité de l'ensemble de ces tracés.

Enfin, maintenant que nous disposons de performances mesurables, nous revenons à l'objectif initial du partitionnement de tracés manuscrits en ligne. Nous avons appliqué notre algorithme de partitionnement à 500 exemples du chiffre 2.

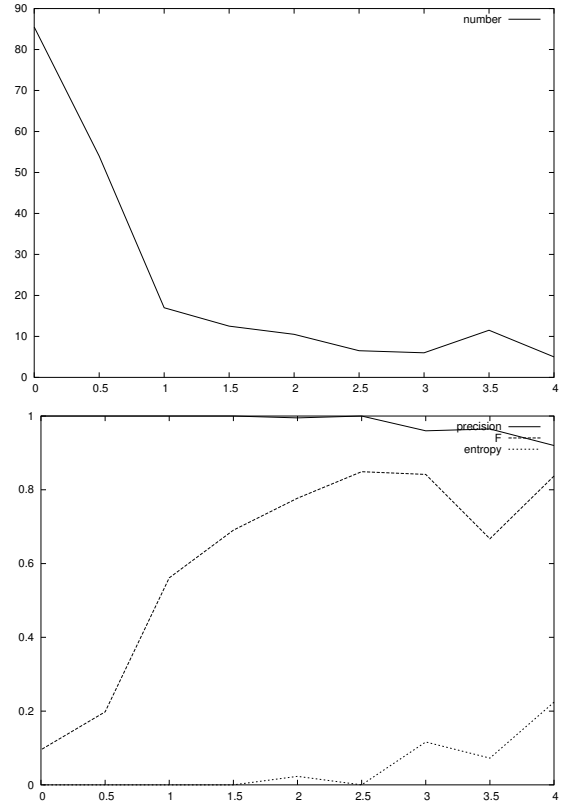

FIG. 4 – Nombre de partitions et mesures de performance en fonction de α dans le cas de séquences avec information de durée

Il n'est pas possible d'obtenir une mesure de performance en l'absence d'évaluations concernant les allographes, mais la figure 5 permet de visualiser graphiquement le partionnement obtenu. Les huit partitions obtenues permettent de distinguer différents allographes du chiffre 2. On reconnaît notamment des allographes caractérisés par soit une forme en 'Z', soit une boucle haute, soit une boucle basse. Les séparations ne sont pas forcément claires, et certains exemples pourraient appartenir à une autre partition.

6 Conclusion

Nous avons présenté un modèle d'identification d'allographes à partir de séquences d'écriture en ligne. Pour cela, nous construisons un modèle MMC, constitué initialement de MMC gauche-droite en parallèle, modélisant chacun une séquence d'apprentissage. Un algorithme de partitionnement utilisant une mesure de similarité spécifique entre MMC nous permet simultanément de déterminer des partitions des séquences (allographes) et d'apprendre des modèles de ces allographes. L'ensemble de notre approche est non-supervisée, et ne fait intervenir aucune information sur les classes des symboles manuscrits, ni sur le nombre de classes présentes dans les données. Les résultats préliminaires sont prometteurs. La visualisation graphique du partitionnement obtenu montre que les résultats sont encourageants et nous travaillons sur la validation de ces résultats sur des bases plus conséquentes.

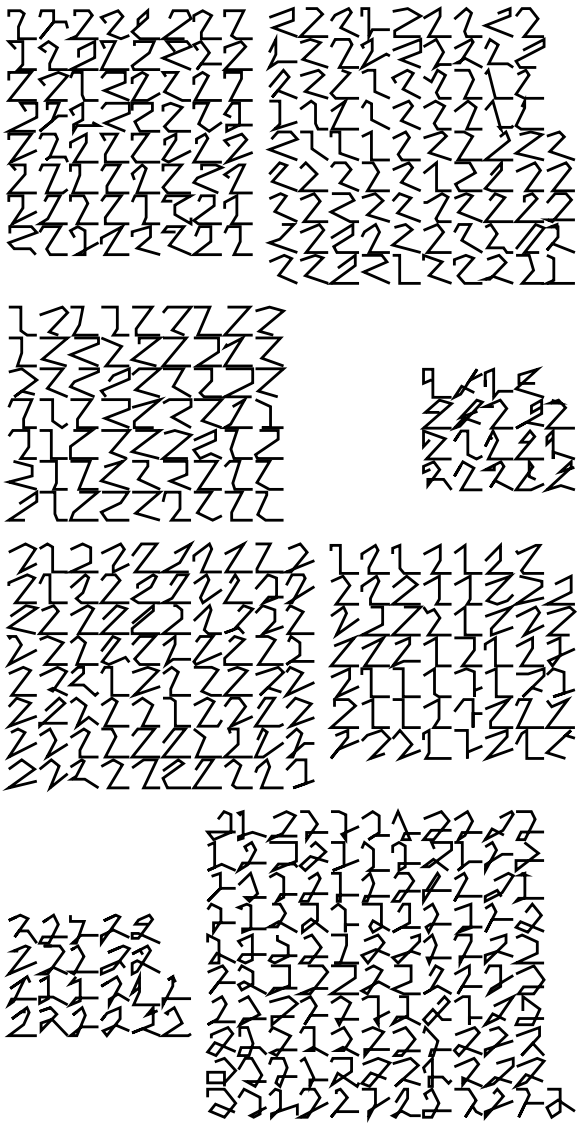

FIG. 5 – Visualisation des allographes du chiffre manuscrit '2' pour $\alpha = 2.5$, correspondant à l'obtention de huit partitions

Références

[ART 02] ARTIÈRES T., GALLINARI P., Stroke level HMMs for on-line handwriting recognition, *8th International Workshop on Frontiers in Handwriting Recognition (IWFHR-8)*, Niagara, août 2002.

[BRA 99] BRAND M., Structure learning in conditional probability models via an entropic prior and parameter extinction, *Neural Computation*, vol. 11, 1999, pp. 1155–1182.

[CAD 00] CADEZ I. V., GAFFNEY S., SMYTH P., A general probabilistic framework for clustering individuals and objects., RAMAKRISHNAN R., STOLFO S., BAYARDO R., PARSA I., Eds., *Proceedinmgs of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-00)*, N. Y., août 20–23 2000, ACM Press, pp. 140–149.

[GUY 94] GUYON I., SCHOMAKER L., PLAMONDON R., LIBERMAN M., JANET S., UNIPEN project of on-line data exchange and benchmarks, *International Conference on Pattern Recognition, ICPR'94*, Jerusalem, Israel, 1994, IEEE Computer Society Press, pp. 29-33.

[LOC 93] LOCKWOOD P., BLANCHET M., An Algorithm For The Dynamic Inference Of Hidden Markov Models (DIHMM), *Proc. ICASSP93*, 1993, pp. 251–254.

[MAR 03] MARUKATAT S., SICARD R., ARTIÈRES T., GALLINARI P., A flexible recognition engine for complex on-line handwritten character recognition, *7th International Conference on Document Analysis and Recognition (ICDAR 2003)*, Edinbourgh, Scotland, août 2003.

[NOS 03] NOSARY A., HEUTTE L., PAQUET T., Unsupervised writer adaption applied to handwritten text recognition, *Pattern Recognition*, vol. 37, 2003, pp. 385-388.

[PER 00] PERRONE M., CONNELL S., K-means clustering for hidden Markov models, *In Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition*, Amsterdam, Netherlands, September 2000, pp. 229–238.

[PRE 00] PREVOST L., MILGRAM M., Modelizing character allographs in omni-scriptor frame : a new non-supervised clustering algorithm, *Pattern Recognition Letters*, vol. 21, n° 4, 2000, pp. 295-302.

[RIS 82] RISSANEN J., A universal prior for integers and estimation by Minimum Description Length, *Annals of Statistics*, vol. 11, 1982, pp. 416–431.

[STE 00] STEINBACH M., KARYPIS G., KUMAR V., A comparison of document clustering techniques, 2000.

[STO 93] STOLCKE A., OMOHUNDRO S., Hidden Markov Model Induction by Bayesian Model Merging, HANSON S. J., COWAN J. D., GILES C. L., Eds., *Advances in Neural Information Processing Systems*, vol. 5, Morgan Kaufmann, San Mateo, CA, 1993, pp. 11–18.

[VUU 97] VUURPIJL L., SCHOMAKER L., Finding structure in diversity : A hierarchical clustering method for the categorization of allographs in handwriting, 1997.