Champs de Markov conditionnels pour le traitement de séquences[1]

Trinh Minh Tri Do*, Thierry Artières*

*LIP6, Université Paris 6
8 rue du capitaine Scott
75015 Paris France
Do@poleia.lip6.fr, Thierry.Artieres@lip6.fr

Résumé. Les modèles conditionnels du type modèles de Markov d'entropie maximale et champs de Markov conditionnels apportent des réponses aux lacunes des modèles de Markov cachés traditionnellement employés pour la classification et la segmentation de séquences. Ces modèles conditionnels ont été essentiellement utilisés jusqu'à présent dans des tâches d'extraction d'information ou d'étiquetage morphosyntaxique. Cette contribution explore l'emploi de ces modèles pour des données de nature différente, de type « signal », telles que la parole ou l'écriture en ligne. Nous proposons des architectures de modèles adaptées à ces tâches pour lesquelles nous avons dérivé les algorithmes d'inférence et d'apprentissage correspondant. Nous fournissons des résultats expérimentaux pour deux tâches de classification et d'étiquetage de séquences.

1 Introduction

La classification, la segmentation et l'étiquetage de données séquentielles sont des problématiques au cœur de nombreux domaines comme la bioinformatique, la reconnaissance de l'écriture, l'extraction d'information. Une des problématiques principales dans ce type de domaine consiste en effet à transformer une séquence observée (un signal écrit par exemple) en une séquence d'étiquettes (on utilise également le terme de labels). Cette tâche peut être réalisée à différents niveaux. On cherche à segmenter le signal écrit d'une phrase en une séquence de mots, de même que le signal écrit de chaque mot doit être segmenté en une séquence de caractères, etc.

Les modèles Markoviens cachés (MMC) constituent l'approche la plus utilisée pour résoudre ce type de tâches bien qu'ils reposent sur des hypothèses d'indépendance fortes sur les données et qu'ils soient appris de façon non discriminante. Ce dernier point vient du fait que ce sont des modèles génératifs et qu'ils définissent une loi de probabilité conjointe $P(X,Y)$ sur la séquence d'observations X et la séquence d'étiquettes associée Y. Diverses

[1] Ce travail est en partie financé par le programme IST de la communauté européenne, à travers le réseau d'Excellence PASCAL IST-2002-506778.

méthodes ont été proposées pour introduire de l'information discriminante dans des systèmes de type Markovien ou plus généralement basés sur des modèles génératifs (Jaakkola et al., 1999, Bahlmann et al., 2002, Moreno et al., 2003, Do et al., 2005). Ces travaux reposent en grande partie sur des méthodes à noyau et des machines à vecteur support.

Des modèles sont apparus récemment qui visent à palier à l'ensemble des défauts des MMC. Ce sont des modèles conditionnels qui s'attachent à modéliser la loi de probabilité conditionnelle $p(Y/X)$. On peut citer les modèles de Markov à entropie maximale (McCallum et al., 2000) et les champs de Markov conditionnels (Lafferty et al., 2001). Ces modèles ont été essentiellement utilisés jusqu'à présent dans le traitement de documents textuels, pour la reconnaissance d'entités nommées, l'extraction d'information ou l'étiquetage morphosyntaxique. Les caractéristiques employées et les algorithmes d'apprentissage sont par conséquent adaptés à ces contextes applicatifs. Cette contribution explore l'emploi de modèles conditionnels pour la reconnaissance de données de type « signal », telles que la parole ou l'écriture en ligne. Plusieurs adaptations sont nécessaires. Tout d'abord, les observations sont de nature différente, ce sont généralement des séquences de vecteurs réels dans R^p. Ensuite, les classes sont souvent multimodales, il y a par exemple plusieurs façons d'écrire un « a ». Enfin, l'étiquetage des données dans la phase d'apprentissage est le plus souvent partiel -- on connaît la classe d'une séquence d'observations (e.g. un « a ») mais on ne connaît pas la séquence d'états correspondante -- alors que les algorithmes proposés dans la littérature requièrent une base de données totalement étiquetée.

Dans la suite, nous commençons par introduire les modèles conditionnels. Puis nous présentons des modèles conditionnels adaptés à des classes multimodales et dérivons les algorithmes pour apprendre ces modèles avec des données partiellement étiquetées. Enfin, nous fournissons des résultats expérimentaux pour deux tâches de classification de séquences, la reconnaissance de caractères manuscrits en ligne et la reconnaissance de comportements de l'utilisateur en se basant sur les mouvements de son œil[2], en comparant modèles conditionnels et modèles Markoviens standards.

2 Modèles conditionnels pour données séquentielles

Les modèles présentés ici sont utilisés pour des tâches de classification ou de segmentation dont le principe est le suivant. L'étiquetage (ou la segmentation) d'une séquence d'observations $X = x_1 ... x_T$ consiste à identifier parmi toutes les segmentations possibles, $Y = y_1, ..., y_T$, la meilleure séquence de labels (ou étiquettes), Y^*, telle que :

$$Y^* = \arg\max_Y P(Y/X) = \arg\max_Y \frac{P(X,Y)}{P(X)} = \arg\max_Y P(X,Y) \quad (1)$$

Dans la suite, on suppose que toutes les composantes y_t de Y parcourent un alphabet fini Ψ. Par exemple, X est l'ensemble des phrases du langage naturel et Y est l'ensemble des étiquetages morpho-syntaxiques de ces phrases (dans ce cas, Ψ est l'ensemble des étiquettes morpho-syntaxiques possibles). Nous commençons par décrire les modèles de

[2] Ces données proviennent du challenge « *Inferring Relevance From Eye Movements Challenge 2005* » organisé par le réseau d'excellence PASCAL.

Markov cachés (MMC) puis les modèles de Markov à entropie maximale (MMEM) (McCallum et al., 2000). Nous présentons ensuite les champs de Markov conditionnels (CMC) (Lafferty et al., 2001) ainsi qu'une extension, les CMC semi-Markoviens (Sarawagi et Cohen, 2005).

2.1 Modèles de Markov cachés (MMC)

Un MMC est défini structurellement par un ensemble fini d'états, *S*, et un espace d'observation, cela peut être R^p dans le cas de modèles dits continus ou bien un alphabet fini $\mathcal{X}$ d'observations (e.g. un dictionnaire de mots) dans le cas de modèles discrets. Un MMC est également défini par ses paramètres : une loi de probabilité pour l'état initial $\{P(s), s \in S\}$, des distributions conditionnelles $\{P(s'/s), (s,s') \in S^2\}$ qui représentent les probabilités de transiter d'un état s vers un état s', et des lois de probabilité conditionnelles des observations $\{P(x/s), s \in S\}$, qui représentent les probabilités d'émission des observations.

Les MMC traditionnellement utilisés font des hypothèses sur les données : la probabilité d'être dans un état à l'instant t ne dépend que de l'état à l'instant précédent, à $t-1$; et l'observation émise à un instant t ne dépend que de l'état à l'instant t. En exploitant ces deux hypothèses, on peut montrer que la probabilité jointe d'une séquence d'états Y (i.e. une segmentation de *X* dans le MMC) et d'une séquence d'observations X est définie par :

$$P(X,Y) = P(y_1)P(x_1/y_1)\prod_{t=2}^{T} P(y_t/y_{t-1})P(x_t/y_t) \quad (2)$$

Les MMC ont deux inconvénients majeurs. D'une part ce sont des modèles génératifs qui définissent une distribution de probabilité jointe $P(X,Y)$ sur la séquence d'observations et de labels. Or ce n'est qu'un moyen détourné d'apprendre un modèle pour la classification de séquences (cf. équation (1)). D'autre part, les hypothèses d'indépendance sous-jacentes à l'emploi de MMC sont fortes et rarement vérifiées.

2.2 Modèles de Markov à entropie maximale (MMEM)

Les modèles de Markov à maximum d'entropie (MMEM) sont des modèles conditionnels qui permettent d'éviter en partie ces limites des modèles génératifs. Ces modèles définissent une loi de probabilité conditionnelle $P(Y/X)$. En supposant que la séquence d'étiquettes obéit à un processus Markovien, on montre que :

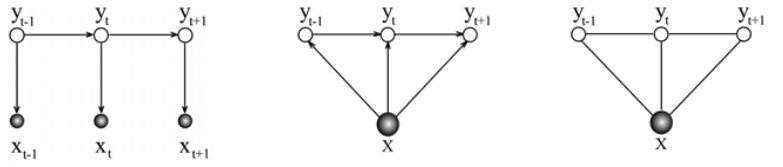
$$P(Y/X) = P(y_1/x_1)\prod_{t=2}^{T} P(y_t/y_{t-1}, X) \quad (3)$$

En comparant à l'équation (2) on voit que l'on a remplacé les probabilité de transition $P(s'/s)$ par des probabilités conditionnelles de la forme $P(s'/s,X)$ et les probabilités d'émission n'apparaissent pas dans cette formule. L'apprentissage est ainsi focalisé sur ce qui différencie une transition d'une autre sans avoir à modéliser complètement le processus de génération des données, i.e. on ne modélise pas la marginale *P(X)*. Les MMEM sont des modèles discriminants. Un autre avantage vient du fait que cette modélisation ne réclame pas d'hypothèses particulières sur *X* et que l'on peut paramétrer les lois de probabilité de

transition par des caractéristiques exploitant des dépendances complexes entre les observations de la séquence *X*. (Lafferty et al., 2001) ont toutefois mis en évidence un comportement indésirable des MMEM qu'ils ont appelé « *label bias* » que nous ne décrivons que succinctement. Ce problème vient du fait que les lois de probabilités de transitions à partir d'un état sont normalisées, c'est à dire que $\sum_{s'} P_s(s'/X) = 1$. Cela induit que si la structure du modèle est telle qu'un état n'a qu'un état successeur, l'observation *X* n'a aucune influence sur le décodage.

2.3 Champs de Markov conditionnels (CMC)

Les champs de Markov conditionnels sont une instance particulière des champs aléatoires. Ces derniers permettent d'estimer des lois de probabilité jointes sur un graphe de nœuds représentant des variables aléatoires. Les champs aléatoires conditionnels font de même mais en conditionnant les valeurs des noeuds représentant les variables à estimer (les labels) par les valeurs de noeuds représentant les variables d'entrée. La figure 1 illustre les différences fondamentales entre les modèles MMC, MMEM et CMC en les représentant sous forme de modèles graphiques. La figure 1 (c) est une représentation graphique d'un champ aléatoire conditionnel (avec une structure de chaîne). Cette représentation est à comparer avec celle d'un MMC (Fig. 1 (a)) et celle d'un MMEM (Fig. 1 (b)). Notons que ces deux derniers modèles sont représentés par des graphes orientés qui permettent d'exprimer les lois de probabilité, jointe ou conditionnelle, suivant les formules (2) ou (3). Notons également que les MMEM et les CMC, étant des modèles conditionnels, ne requièrent pas d'hypothèses particulières sur *X*, ce qui explique les nœuds non décomposés *X* dans les figures 2 (b) et (c).



(a) (b) (c)

FIG. 1 - *Représentation de MMC (a), MMEM (b) et CMC (c) sous forme de modèles graphiques, les nœuds grisés correspondent aux variables observées.*

Nous considérons ici des champs aléatoires --définis par un graphe de nœuds et de liens-- tels que, conditionnellement à *X*, la séquence d'étiquettes *Y* obéit à la propriété Markovienne exprimée par le graphe des nœuds. C'est à dire que $P(y_t / X, \{y_{t'}, t \neq t'\}) = P(y_t / X, \{y_{t'}, V(t,t')\})$, où $V(t,t')$ signifie que *t* et *t'* sont des voisins dans le graphe. En exploitant la théorie des champs aléatoires (Lafferty 2001), on peut montrer que la probabilité conditionnelle d'une séquence d'étiquettes *Y* connaissant l'observation *X* peut se mettre sous la forme :

$$P(Y/X,W) = \frac{score(X,Y,W)}{Z_W(X)} = \frac{e^{W.F(X,Y)}}{\sum_{Y'} e^{W.F(X,Y')}} \quad (4)$$

Où $Z_W(X) = \sum_{Y'} e^{W.F(X,Y')}$ est un facteur de normalisation et $F(X,Y)$ est un vecteur de caractéristiques calculées en fonction de l'observation *X* et de la séquence d'états *Y*. *W* est un vecteur de poids. La loi de probabilité conditionnelle s'exprime donc en fonction de caractéristiques (rassemblées dans le vecteur $F(X,Y)$) qu'il faut définir explicitement. La théorie des champs de Markov aléatoires dit que ces caractéristiques sont définies sur les cliques maximales du graphe. Dans le cas d'une structure en chaîne (cf. Fig. 1 (c)), les caractéristiques sont définies soit sur un état, soit sur deux états successifs. Soit $f = < f_1,..., f_L >$ l'ensemble des fonctions caractéristiques. Alors :

$$F(X,Y) = \sum_{t=1}^{T} f(t,X,Y) = \left\langle \sum_{t=1}^{T} f_1(t,X,Y), \sum_{t=1}^{T} f_2(t,X,Y),..., \sum_{t=1}^{T} f_L(t,X,Y) \right\rangle \quad (5)$$

Des caractéristiques typiquement utilisées dans des problèmes d'étiquetage morpho syntaxiques sont par exemple « le mot commence par une majuscule ». Dans le cas de données comme la parole ou l'écriture en ligne on ne peut pas définir des caractéristiques de haut niveau de ce type. Dans ces domaines, un signal d'entrée est transformé après prétraitement en une séquence de vecteurs de *p* caractéristiques réelles. On utilisera alors ces *p* caractéristiques dans les CMC.

L'inférence est réalisée à l'aide d'un algorithme de programmation dynamique. Si la topologie du graphe est une chaîne on utilise l'algorithme de Viterbi, si c'est un arbre on peut utiliser l'algorithme Belief Propagation (Weiss, 2001) et pour un graphe quelconque on utilise Loopy Belief Propagation (Pearl, 1988, Murphy et al., 1999). En apprentissage, on dispose d'une base d'apprentissage de *K* exemples complètement étiquetés, $BA = \{(X_k, Y_k)\}_{k=1}^{K}$, où X_k est une séquence d'observations et Y_k est la séquence des étiquettes (i.e. nœuds) correspondant à X_k. Durant l'apprentissage, on cherche les paramètres *W* qui maximisent la log-vraisemblance conditionnelle sur l'ensemble d'apprentissage. Ce critère est convexe et peut être maximisé par une méthode de gradient. Il s'écrit :

$$L(W) = \sum_{k=1}^{K} \log P(Y_k / X_k, W) = \sum_{k=1}^{K} ((W.F(X_k, Y_k) - \log Z_W(X_k)) \quad (6)$$

Il faut noter que le facteur de normalisation $Z_W(X)$ implique une somme sur un nombre exponentiel de séquences d'états possibles, mais cette somme peut être calculée efficacement via un algorithme de programmation dynamique.

2.4 Champs de Markov conditionnels semi-Markoviens (SCMC)

Les CMC présentés précédemment permettent de déterminer la séquence de labels $Y = y_1, y_2,..., y_T$ de probabilité maximale pour une séquence d'observations donnée $X = x_1, x_2,..., x_T$. Cela correspond bien à des problématiques comme l'étiquetage morpho syntaxique, où l'on cherche une étiquette pour chaque mot. Cependant, on peut souhaiter travailler sur des données séquentielles telles que les observations prises isolément ont moins de sens que des segments de plusieurs observations successives. Afin d'intégrer des informations segmentales, (Sarawagi et Cohen, 2004) ont proposé une extension des CMC appelée CMC semi-Markoviens ou semi-CMC (SCMC). Etant donnée une observation $X = x_1, x_2,..., x_T$, on cherche la meilleure segmentation de *X* en étiquetant les

segments. La sortie du SCMC est une séquence $S = s_1, s_2, ..., s_J$, avec $J \leq T$, et $s_j = \langle t_j, u_j, y_j \rangle$ désigne un segment défini par une position de début t_j, une position de fin u_j, et une étiquette (un état) y_j. Dans les SCMC les caractéristiques sont donc calculées par segment plutôt qu'au niveau des éléments individuels. Dans le cas où le graphe a une structure de chaîne, une caractéristique du segment *j*, par exemple la $v^{ème}$ caractéristique du segment *j*, $f_v(j, X, S)$, est calculée à partir de $x_{t_j}, ..., x_{u_j}$, y_j et y_{j-1} uniquement.

L'apprentissage vise, comme dans le cas de CMC, à maximiser le critère de log-vraisemblance conditionnelle. Ici encore, l'algorithme d'apprentissage proposé exploite une base d'apprentissage complètement étiquetée.

3 CMC pour la classification de séquences

3.1 SCMC pour la classification de données monomodales

Les CMC sont, comme nous l'avons déjà mentionné, traditionnellement utilisés pour l'étiquetage de données séquentielles. Pour concevoir un système de classification de séquences basé sur un CMC on peut utiliser une architecture basée sur une structure de chaîne pour chaque classe. La structure de chaîne est en effet naturelle pour modéliser des séquences d'une même classe dans lesquelles on veut distinguer les différentes parties comme le début, le milieu, la fin. Dans le cas de la reconnaissance de l'écriture manuscrite en ligne par exemple, on utilise un modèle de type chaîne pour chaque lettre. Dans la chaîne correspondant au caractère « a », le premier noeud correspond au début du tracé du « a », le second noeud à une partie intermédiaire, etc. On utilise alors une architecture du type mélange de structures en chaîne, telle que celle illustrée figure 2. Il y a une « branche » par classe. Les branches peuvent avoir des nombres d'états différents, il s'agit d'un choix a priori sur la structure du modèle. Bien entendu, la représentation « dynamique » de ce modèle est similaire à la Figure 1 (c), la différence vient du fait que toutes les transitions entre nœuds ne sont pas autorisées. Lorsque l'on apprend un modèle de ce type avec des données des *N* classes, on apprend des paramètres *W* qui permettent de discriminer au mieux entre les séquences d'observations des différentes classes.

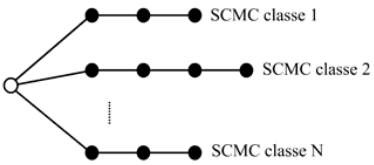


FIG. 2 - *Un champ aléatoire pour la classification de séquences dans le cas de données monomodales. Chaque « branche » correspond à une classe.*

L'étiquetage disponible dans des bases d'apprentissage pour la classification de signaux est souvent minime et réduit à la classe. Cela signifie que l'on sait que la segmentation correspondant à une séquence d'observations correspond à une branche particulière (e.g.

celle du « a ») mais on ne connaît pas la segmentation plus précisément. Or les algorithmes d'apprentissage des CMC requièrent cette information. Nous montrons ici comment apprendre un CMC sans disposer de cette information.

On dispose donc d'une base d'apprentissage de *K* exemples, $BA = \{(X_k, Y_k)\}_{k=1}^{K}$, où X_k est une séquence d'observations et Y_k est la classe correspondant à X_k. Alors, la probabilité conditionnelle $P(Y_k / X_k)$ peut s'écrire :

$$P(Y_k / X_k) = \sum_{S \in S(Y_k)} P(Y_k, S / X_k) = \frac{\sum_{S \in S(Y_k)} e^{W.F(S, X_k)}}{Z_W(X_k)} \quad (7)$$

Où *S* désigne une segmentation, $S(Y_k)$ désigne l'ensemble des segmentations pour une séquence de classe Y_k, et $Z_W(X_k)$ est un facteur de normalisation. Cette modélisation ressemble à un mélange de modèles (Quattoni et al., 2004). Pour limiter les problèmes numériques et simplifier l'implémentation on peut choisir d'approximer la quantité précédente par :

$$P(Y_k / X_k) = \sum_{S} P(Y_k, S / X_k) = \frac{\max_{S} e^{W.F(S, X_k)}}{Z_W(X_k)} \quad (8)$$

En apprentissage, on estime les paramètres maximisant la log-vraisemblance conditionnelle, la segmentation (non disponible) des séquences d'apprentissage est donc apprise au fur et à mesure pendant l'apprentissage (Do, 2005). Il faut noter qu'à cause de ces variables cachées, le critère a plusieurs maximums locaux, l'optimisation n'assure donc pas de trouver l'optimum global.

3.2 SCMC pour données multimodales

Bien souvent les classes sont multimodales et la modélisation précédente peut s'avérer insuffisante. Nous proposons dans ce cas des modèles que l'on peut voir comme des mélanges de champs aléatoires conditionnels. Plutôt que d'avoir une structure en chaîne pour chaque classe, on en considère plusieurs, chacune pouvant se spécialiser, pendant l'apprentissage, sur une modalité de la classe.

La difficulté de l'apprentissage consiste ici encore dans le manque d'étiquetage des données en apprentissage. Ici, on dispose de la même information que précédemment pour une séquence d'apprentissage, sa classe. On sait par exemple qu'une séquence d'observations est un « a » mais on ne connaît pas l'allographe (i.e. la branche) correspondant ni la segmentation dans cette branche. On introduit alors un deuxième type de variable cachée, l'indicateur de la branche. On obtient donc :

$$P(Y_k / X_k) = \sum_{B \in B(Y_k)} \left[\sum_{S \in S(Y_k)} P(Y_k, B, S / X_k) \right] = \frac{\sum_{B \in B(Y_k)} \left[\sum_{S \in S(Y_k)} e^{W.F(B, S, X_k)} \right]}{Z_W(X_k)} \quad (9)$$

Où *B* désigne une branche, $B(Y_k)$ désigne l'ensemble des branches correspondant à la classe $B(Y_k)$, et S, $S(Y_k)$ et $Z_W(X_k)$ ont la même signification que précédemment.

Pour simplifier les calculs et l'implémentation on peut choisir d'approximer la quantité précédente en approximant la somme au numérateur par un maximum, comme dans

l'équation (8). En apprentissage, on estime les paramètres maximisant la log-vraisemblance conditionnelle, la segmentation (non disponible) des séquences d'apprentissage est donc apprise au fur et à mesure pendant l'apprentissage. Il faut noter qu'à cause des variables cachées, l'optimisation n'assure pas de trouver l'optimum global.

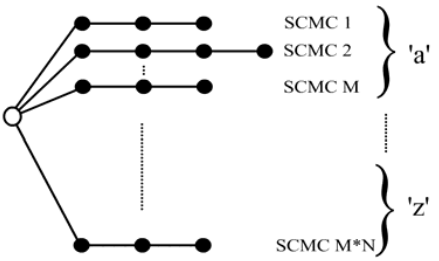


FIG. 3 - *Modèle SCMC avec plusieurs modèles par classe pour prendre en compte les allographes.*

4 Mises en œuvre expérimentales

Les implémentations des CMC ont été réalisées en exploitant une bibliothèque implémentant un algorithme de gradient avec la méthode de quasi-Newton avec mémoire limitée LBFGS (Nocedal, 1989).

4.1 Déterminer l'intérêt d'un utilisateur par des mouvements de l'œil

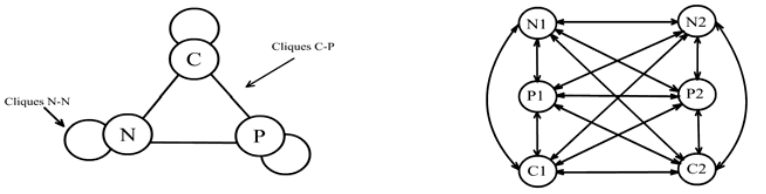
L'étude réalisée dans cette partie a été réalisée dans le cadre d'un challenge organisé par des membres du réseau d'excellence PASCAL. Ce challenge consiste à explorer la possibilité d'inférer l'intérêt d'un utilisateur dans les différentes lignes de texte qui lui sont présentées à l'écran (typiquement retournées par un moteur de recherches) en fonction des mouvements de son œil (Salojärvi et al., 2005). Les données ont été collectées de la façon suivante. Pour une requête donnée, on présente à l'utilisateur 10 titres (un titre est une réponse tenant sur une ligne) correspondant à 10 réponses possibles. Parmi ces 10 titres, un seul est correct, 4 sont pertinents, et 5 sont non pertinents. On observe les mouvements de l'œil de l'utilisateur qui cherche parmi les 10 titres celui qui le satisfait. Une telle expérience (requête + 10 titres de réponses) est appelée dans la suite un assignement.

La trajectoire de l'œil est segmentée en fixations et saccades, puis on détermine à quels mots de la page ces fixations correspondent. Cette séquence ordonnée temporellement de fixations est transformée en une séquence de vecteurs de caractéristiques, un par fixation. On dispose d'une vingtaine de caractéristiques, fournies par les organisateurs du challenge (Salojärvi et al., 2005). On dispose également pour chaque fixation, du titre auquel appartient le mot fixé, de la position du mot dans le titre et de la longueur (en mots) du titre.

Puisque la segmentation en titres est connue, en apprentissage ou en reconnaissance, nous avons calculé des caractéristiques segmentales par segment de fixations correspondant à un même titre. Les vecteurs de caractéristiques sont additionnés sur tout le segment. Dans la

suite, on considère qu'un assignement est représenté par une séquence de vecteurs de 22 caractéristiques $X = x_1...x_J$, où x_j représente la somme des vecteurs caractéristiques correspondant aux visites des mots du $j^{ème}$ titre visité. On cherche à étiqueter cette séquence par une segmentation $Y = y_1,..., y_J$, où $y_j \in \{N, P, C\}$, *N*, *P* et *C* correspondent aux étiquettes *Non Pertinent*, *Pertinent* et *Correct*. Il faut noter que si x_i et x_j sont deux visites d'un même titre (on peut en effet faire des allers et retours sur les titres et donc « visiter » un même titre plusieurs fois) alors ils partagent la même étiquette de classe $y_i = y_j$.

Nous avons utilisé divers modèles CMC. Le premier est illustré par la figure 4 (a), c'est un modèle simple à trois nœuds (on ne représente pas ici les nœuds correspondant à la séquence d'observations), un pour chaque classe. Dans ce graphe, on peut considérer deux types de caractéristiques, des caractéristiques locales et des caractéristiques de transition, reliant deux visites consécutives. Par exemple, on peut avoir des caractéristiques locales au nœud *N*, cela permet de prendre en compte des valeurs absolues de caractéristiques telles que la durée des fixations sur les mots d'un titre non pertinent. On utilise également des caractéristiques de transitions, qui correspondent à la différence des caractéristiques entre les deux visites successives ; cela permet par exemple de prendre en compte la différence de durée entre les fixations d'un titre pertinent et celles d'un titre non pertinent. Ainsi, en notant f_1, …, f_{22} les attributs calculés sur les mots et sommées ensuite sur les segments, les caractéristiques employées dans les CMC sont les attributs bruts en tant que caractéristiques locales et la différence des attributs en tant que caractéristiques de transition.



(a) (b)

FIG. 4 - *Représentation graphique d'un modèle CMC à 3 nœuds (a) et d'un modèle à 6 états (b), dans lequel on distingue une première visite des visites ultérieurs d'un titre.*

Lorsqu'un utilisateur visite une seconde, ou une troisième fois un même titre, on peut supposer qu'il ne le visite pas de la même façon et que cela est informatif. Nous avons donc également utilisé des modèles dans lesquels on distingue la première visite d'un titre de la classe *N* (ou *P* ou *C*) d'une deuxième visite d'un tel titre. Pour cela, il suffit de multiplier les nœuds (Figure 4 (b)). Le dernier titre visité est souvent très important, car c'est souvent le titre correct. Pour le prendre en compte automatiquement, nous avons rajouté trois nœuds correspondant à la dernière visite. Le décodage consiste à tester parmi toutes les possibilités (étiquetage des 10 titres parmi *N*, *P* ou *C*) celle de probabilité maximale.

Nous avons comparé ces systèmes conditionnels à des systèmes Markovien plus standard. Pour cela, nous avons appris des MMC similaires à 3, 6 ou 9 états dans lesquels les probabilités d'émission sont modélisées par des lois gaussiennes sur les vecteurs de

caractéristiques. Il y a 336 assignements dans la base d'apprentissage et 149 assignements dans la base de test. Les performances sont calculées suivant la procédure proposée dans (Salojärvi et al., 2005), c'est à dire que l'on n'évalue le système que sur les réponses qu'il fournit pour les titres qui ont été effectivement visités. Le tableau 1 montre les performances des systèmes génératifs (MMC) et des modèles conditionnels (CMC). Les meilleurs résultats ont été obtenus avec des MMC à 6 états alors que les SCMC ont montré des performances supérieures quel que soit le nombre d'états, et allant jusqu'à 73% lorsque l'on distingue la première visite, la dernière visite et les visites intermédiaires à un même titre.

Méthode	Performance
MMC 6 états	66.2
SCMC 3 états	71,0
SCMC 6 états	71.8
SCMC 9 états	73,2

TAB. 1 - *Performances du meilleur système Markovien (MMC) et de systèmes SCMC pour l'étiquetage des titres en fonction des mouvements de l'œil.*

4.2 Reconnaissance de l'écriture manuscrite en ligne

Un signal d'écriture manuscrite en ligne est un signal temporel constitué des coordonnées successives d'un stylo, il est capturé sur une tablette digitale ou via un stylo électronique. La base UNIPEN (Guyon et al., 1994) est une base internationale de référence dans le domaine de la reconnaissance d'écriture. Nous avons travaillé sur une partie de cette base regroupant des signaux de 200 scripteurs et correspondant aux 26 caractères. Notre base contient environ 60000 exemples, nous en utilisons 33% pour l'apprentissage et 66% pour le test.

Le signal d'écriture étant très variable (il existe de nombreux allographes pour tracer un même caractère) l'usage de modèles de mélanges ou de systèmes basés sur des prototypes de tracés typiques est extrêmement répandu. Les systèmes les plus performants sont souvent à base de MMC. L'apprentissage de ce type de modèles n'est pas aisé car le nombre d'allographes ainsi que la topologie des modèles Markoviens les modélisant doivent être fixés à la main. Divers travaux ont été menés pour apprendre complètement les modèles de caractères à partir des données (Lee et al., 2001, Artières et Gallinari 2002), ils sont basés sur la construction d'un MMC à partir d'un seul tracé et utilisent une représentation des tracés sous forme de séquence de codes directionnels. Le système de référence est un système de ce type, dont les détails peuvent être trouvés dans (Marukatat, 2004). Un des intérêts du système de référence Markovien réside dans sa capacité à apprendre la topologie (nombre de branches, nombre d'états) des modèles à partir des données, ce que nous ne savons pas faire aujourd'hui dans le cas de CMC.

Nous avons donc mis en œuvre des CMC en fixant leur topologie d'après celle apprise par le système Markovien. L'idée consiste à construire un modèle CMC multi branches en reprenant les topologies des modèles MMC appris. De plus, on utilisera dans ce CMC les mêmes caractéristiques que celles utilisées dans les MMC. Le tableau 2 résume les performances du système Markovien et de systèmes SCMC pour la classification des 26 caractères minuscules. Les performances s'améliorent avec la taille des modèles. Comme on le voit, les SCMC surpassent le système Markovien dans toutes les expériences.

Nombre de modalités (branches) par caractère	MMC	SCMC
1	67.4%	76.4%
3	79.3%	84.6%
5	81.6%	85.9%
10	84.4%	87.6%

TAB. 2 - *Performance des systèmes pour la classification des caractères minuscules.*

On voit ici encore que les systèmes à base de CMC surpassent dans tous les cas le système Markovien de référence, ce qui est très intéressant. Car ce dernier système a fait l'objet de nombreux travaux depuis quelques années dans notre équipe et est au niveau de l'état de l'art dans le domaine. Il reste néanmoins du travail à réaliser pour mettre en œuvre des CMC. Les algorithmes d'apprentissage sont encore relativement coûteux. Et surtout, la structure des CMC est déterminée d'après la structure apprise par le système Markovien.

5 Conclusion

Nous avons exploré dans cette contribution l'emploi de champs aléatoires Markoviens conditionnels pour la classification et l'étiquetage de signaux. Tout d'abord, nous nous sommes focalisés sur des variantes des CMC exploitant la notion de segments et des caractéristiques segmentales. Nous avons développé des algorithmes pour réaliser l'apprentissage de nos modèles en présence d'une information de segmentation minimale, de type classe. Nous avons fourni des résultats expérimentaux montrant que ces modèles surpassent des modèles Markoviens plus traditionnels dans deux tâches très différentes de classification de données séquentielles.

Références

Artières T. et P. Gallinari (2002). *Stroke level HMMs for on-line handwriting recognition*, International Workshop on Frontiers in Handwriting Recognition.

Bahlmann C., B. Haasdonk, H. Burkhardt (2002). *On-line Handwriting Recognition using Support Vector Machines - A kernel approach*. International Workshop on Frontiers in Handwriting Recognition.

Nocedal J. and D.C. Liu (1989). *On the limited memory BFGS method for large-scale optimization*. Mathematic Programming, 45:503-528.

Do T.M.T. (2005). *Champs de Markov conditionnels pour le traitement de séquences*. Rapport de stage, Master Recherche, Université Paris 6, Septembre 2005.

Do T.M.T., T. Artières , P. Gallinari (2005). *Sélection de Modèles par des Méthodes à Noyaux pour la classification de données séquentielles*. In EGC 2005.

Guyon I., L. Schomaker , R. Plamondon , M. Liberman , S. Janet (1994). *UNIPEN project of on-line data exchange and recognizer benchmark*. International Conference on Pattern Recognition.

Jaakkola T., M. Diekhans, D. Haussler (1999). *Using the Fisher kernel method to detect remote protein homologies*. International Conference on Intelligent Systems for Molecular Biology.

Lafferty J., A. McCallum, and F. Pereira (2001). *Conditional random fields: Probabilistic models for segmenting and labeling sequence data.* International Conf. on Machine Learning, 282–289. Morgan Kaufmann, San Francisco, CA.

Lee J.J., J. Kim and J.H. Kim (2001). *Data-driven design of HMM toplogy for on-line handwriting recognition.* International Journal of Pattern Recognition and Artificial Intelligence, Vol. 15, n° 1, pp 107-121.

Marukatat S. (2004) *Une approche générique pour la reconnaissance de signaux écrits en ligne*. Thèse de doctorat, Université Paris 6, LIP6.

McCallum, A., D. Freitag, and F. Pereira (2000) *Maximum entropy Markov models for information extraction and segmentation*. In Proc. ICML.

Moreno P.J., P.P. Ho, N. Vasconcelos (2003). *A Generative Model Based Kernel for SVM classification in Multimedia applications*. NIPS.

Murphy K., Y. Weiss and M. Jordan (1999). *Loopy belief propagation for approximate inference: an empirical study*. In Proc. of the Conf. on Uncertainty in AI.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Quattoni A., M. Collins and T. Darrel (2004). *Conditional Random Fields for Object Recognition*. In Advances in Neural Information Processing Systems 17, 2004.

Salojärvi J., K. Puolamäki, J. Simola, L. Kovanen, I. Kojo, S. Kaski (2005). *Inferring Relevance from Eye Movements: Feature Extraction.* Helsinki University of Technology, Publications in Computer and Information Science, Report A82.

Sarawagi S. and W. Cohen (2004). *Semi-Markov Conditional Random Fields for Information Extraction*. Advances in Neural Information Processing Systems.

Weiss Y. (2001). *Correctness of belief propagation in Gaussian graphical models of arbitrary topology*. Neural Computation, 13:2173-2200.

Summary

Maximum Entropy Markov Models and Conditional Random Fields have been designed to address the limits of hidden Markov models traditionally employed for the classification and the segmentation of sequences. These conditional models have mainly been introduced up to now in information retrieval or POS-tagging tasks. This paper investigates the use of these models for data of different nature, such as speech or online handwriting. We propose some architectures adapted to such tasks and derive inference and training algorithms. We provide experimental results for two classification and labelling tasks.