



**HAL**  
open science

# Champs Aléatoires Conditionnels et fonctions de caractéristique à quantification multi-échelle Application à l'extraction de structures dans des journaux d'archive

David Hebert, Thierry Paquet, Stephane Nicolas

► **To cite this version:**

David Hebert, Thierry Paquet, Stephane Nicolas. Champs Aléatoires Conditionnels et fonctions de caractéristique à quantification multi-échelle Application à l'extraction de structures dans des journaux d'archive. RFIA 2012 (Reconnaissance des Formes et Intelligence Artificielle), Jan 2012, Lyon, France. pp.978-2-9539515-2-3. hal-00656534

**HAL Id: hal-00656534**

**<https://hal.science/hal-00656534>**

Submitted on 17 Jan 2012

**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.

# Champs Aléatoires Conditionnels et fonctions de caractéristiques à quantification multi-échelle

## Application à l'extraction de structures dans des journaux d'archive

D. Hébert

T. Paquet

S. Nicolas

Laboratoire LITIS EA 4108  
Université de Rouen, France  
Prénom.Nom@univ-rouen.fr

### Résumé

Nous présentons ici le formalisme de fonctions de caractéristiques quantifiées pour transformer des données continues ou discrètes à valeurs dans de grands intervalles, en données compatibles avec la représentation symbolique utilisée par les champs aléatoires conditionnels (CAC). Nous montrons qu'une conversion simple des données permet au CAC de sélectionner automatiquement les caractéristiques discriminantes qui permettent d'obtenir de meilleures performances. Ce système est évalué sur une tâche de segmentation d'images dégradées de journaux d'archives. Les résultats obtenus montrent la capacité du modèle CAC à travailler avec des données numériques de manière similaire à l'utilisation d'une représentation symbolique, grâce à l'utilisation des fonctions de caractéristiques quantifiées. La tâche de segmentation est réalisée par la définition d'un modèle CAC horizontal dédié à l'étiquetage de pixels.

### Mots Clef

Champs Aléatoires Conditionnels linéaires, fonctions de caractéristiques quantifiées, étiquetage d'images de documents

### Abstract

We introduce quantization feature functions to represent continuous or large range discrete data into the symbolic CRF data representation. We show that doing this conversion in a simple way allows the CRF to automatically select discriminative features to achieve best performance. This system is evaluated on a segmentation task of degraded newspapers archives. The results obtained show the ability of the CRF model to deal with numerical features similarly as for symbolic representation thanks to the use of quantization feature functions. The segmentation task is achieved by the definition of a horizontal CRF model dedicated to pixel labelling.

### Keywords

L-CRF, multi-scale quantization feature functions, document images labelling

## 1 Introduction

Dans cet article, nous explorons l'utilisation des champs aléatoires conditionnels pour l'étiquetage de données continues (ou discrètes à valeurs dans de grands intervalles) et plus particulièrement pour la segmentation d'images de documents en entités fonctionnelles comme des titres, des séparateurs graphiques, des lignes de texte ou des colonnes. L'expérimentation s'appuie sur une tâche de détection d'articles dans des journaux anciens dont la mise en page évolue sur une période de presque 180 ans.

Il est maintenant bien connu qu'une tâche de segmentation difficile doit intégrer un étage de reconnaissance pour améliorer les résultats. Cette affirmation est particulièrement vraie dans le cadre de la reconnaissance de l'écriture manuscrite cursive [8] [9], où réaliser les étapes de segmentation et de reconnaissance conjointement permet d'augmenter significativement la qualité des résultats. Des schémas similaires ont également été proposés pour l'analyse d'images et notamment pour la détection d'objets dans des scènes naturelles [5]. Des travaux ont également été proposés pour la segmentation de documents manuscrits [10].

Les champs aléatoires conditionnels -CAC- (Conditional Random Fields -CRF- dans la littérature anglaise) présentés en 2001 par Lafferty et al. [2] ont ouvert de nouvelles portes pour l'analyse de séquences. Jusque là, les modèles de Markov cachés (MMC) étaient la méthode la plus utilisée pour ce type d'analyse et différentes approches dans différents domaines ont été proposées : détection, segmentation, classification... Comme pour les MMC, les CAC ont aussi été utilisés pour segmenter des images.

Dans sa définition originale, un CAC est un processus stochastique qui modélise les dépendances entre un ensemble d'observations discrètes réalisées sur une séquence discrète (à l'origine, une séquence de mots) et un ensemble d'étiquettes (analyse morphosyntaxique). En comparaison avec un MMC, un CAC ne repose pas sur l'hypothèse forte d'indépendance des observations entre elles conditionnellement aux états associés. Un autre avantage du CAC

face au MMC est qu'il ne combine pas des probabilités conditionnelles locales, évitant ainsi une estimation biaisée de ces probabilités si trop peu d'exemples étiquetés sont disponibles. Le CAC combine des potentiels (positifs ou négatifs) pour prendre en compte des contributions positives ou négatives des observations dans le choix des étiquettes.

L'utilisation d'un CAC pour segmenter une image demande quelques adaptations. Généralement, il s'agit d'une étape de pré-traitement permettant au CAC de travailler avec des données discrètes à partir de données extraites de l'image. Ainsi, He et al. [5] proposent un système basé sur un CAC multi-couches pour effectuer une segmentation d'images naturelles et utilisent les sorties d'un réseau de neurones comme entrées du CAC. Dans [11], un SVM (support vector machine) est utilisé pour modéliser les dépendances entre un pixel et son étiquette. Les CAC ont également été utilisés dans divers travaux dans le cadre de l'extraction de la structure d'un document. Parmi ces travaux, nous pouvons citer Nicolas et al. [10] qui utilisent une approche basée sur un CAC 2D mais également combiné avec un MLP (Multi Layer Perceptron). Un autre exemple est donnée dans [12] où le CAC est explicitement défini comme un second système après une étape de classification de pixels.

Contrairement à ces travaux qui utilisent un étage de discrétisation en amont d'un CAC, nous proposons d'utiliser le formalisme de fonctions de caractéristiques quantifiées directement optimisées lors de la phase d'entraînement du système, permettant ainsi d'effectuer un entraînement efficace en une étape pour un CAC utilisant des données continues.

La suite de cet article est organisée de la manière suivante. Dans la section 2 nous rappellerons brièvement la définition originale d'un modèle CAC linéaire. La section 3 décrit les fonctions de caractéristiques quantifiées que nous proposons. La configuration de notre expérimentation sur la segmentation d'images de journaux anciens est présentée dans la section 4. Les résultats de cette expérimentation sont donnés dans la section 5. Enfin, nous discuterons les résultats obtenus et présenterons une conclusion sur ces travaux en section 6.

## 2 CAC linéaire

Le L-CAC, pour champ aléatoire conditionnel linéaire, a été présenté pour la première fois par Lafferty et al. en 2001 [2]. Les auteurs présentent un modèle discriminant qui ne fait pas l'hypothèse qu'une observation conditionnée par son étiquette est indépendante des observations voisines. Ce modèle est présenté comme une solution au problème de biais de label dont souffrent les MEMM (Maximum Entropy Markov Model) ou les MMC pour l'analyse automatique de la langue.

### 2.1 Le modèle CAC

En nous appuyant sur [7], nous rappelons les propriétés principales d'un CAC. Nous définissons quelques notations pour la compréhension de la suite de cet article.

$X = x_1, x_2, \dots, x_T$  est une séquence de  $T$  observations discrètes.

$Y = y_1, y_2, \dots, y_T$  est la séquence des  $T$  étiquettes associées aux observations de la séquence  $X$

$L$  est l'ensemble des étiquettes possibles (les valeurs possibles pour les  $y_t$ )

$O$  est l'ensemble des observations (les valeurs possibles pour les  $x_t$ , par exemple un lexique discret si les observations sont des mots)

Un L-CAC est défini par :

$$p(Y|X) = \frac{1}{Z(X)} \exp \left( \sum_t \sum_k \lambda_k f_k(y_{t-1}, y_t, x, t) \right)$$

Comme nous pouvons le voir, la probabilité qu'une séquence d'étiquettes particulière  $Y$  soit associée à la séquence d'observations  $X$  est obtenue par une combinaison linéaire de poids  $\lambda_k$  associés à des fonctions  $f_k$  sur la séquence d'observations. Dans le cas d'un CAC discret, ces fonctions sont généralement binaires. Les poids (ou potentiels)  $\lambda_k$  sont les paramètres du modèle et peuvent être interprétés comme l'importance ou la fiabilité de l'information apportée par la fonction binaire  $f_k$ .  $f_k(y_{t-1}, y_t, x, t)$  est la notation générale des fonctions  $f_k$  appelées fonctions de caractéristiques, qui rendent compte chacune de l'occurrence d'une combinaison d'observation(s) et de label(s) particulière.

Par exemple,

$$f_k(y = l_i, x = o_j) = \begin{cases} 1 & \text{si } y_t = l_i \text{ et } x_t = o_j \\ 0 & \text{sinon} \end{cases}$$

Les fonctions de caractéristiques sont définies par l'utilisateur. Elles reflètent la connaissance de l'utilisateur dans le domaine d'application. En ne faisant pas d'hypothèse d'indépendance des observations entre elles conditionnellement à leurs étiquettes, le modèle CAC permet de définir tout un ensemble de caractéristiques contextuelles, ce qui n'est pas possible avec un MMC traditionnel. Ces fonctions de caractéristiques contextuelles sont définies à l'aide de modèles de combinaison contextuels (ou pattern dans la littérature anglophone) entre observation(s) et étiquettes(s). Un modèle de combinaison définit une combinaison contextuelle d'une ou plusieurs observations et d'une ou plusieurs étiquettes. Par exemple, le modèle de combinaison contextuel  $f(y_t, x_t)$  va prendre en compte tous les couples (étiquette, observation) à chaque position  $t$  dans la séquence. Ce modèle peut générer au maximum  $Card(O) \times Card(L)$  fonctions de caractéristiques binaires.

### 2.2 Entraînement d'un L-CAC

Entraîner un CAC signifie optimiser ses paramètres  $\lambda_k$  sur un ensemble d'exemples et de leur vérité terrain. L'idée est

de maximiser la vraisemblance sur un ensemble de données d'entraînement composé de  $N$  couples séquences d'observations/séquences d'étiquettes :

$$\ell(\theta) = \sum_{i=1}^N \log p(y^{(i)}|x^{(i)}; \theta) \text{ avec } \theta = \{\lambda_k\}$$

Optimiser la valeur de  $\ell(\theta)$  est un problème d'optimisation convexe n'ayant qu'un seul optimum global, puisque le modèle est linéaire en fonction des paramètres  $\lambda_k$ . Cette propriété est intéressante car elle laisse un vaste choix d'algorithmes d'optimisation capables de résoudre un tel problème. Cependant, le nombre de fonctions de caractéristiques (et donc de paramètres  $\lambda_k$ ) augmente très rapidement en fonction de la taille de l'ensemble des observations  $O$ , de la taille de l'ensemble des étiquettes  $L$  mais aussi du nombre de modèles de combinaison contextuelle utilisés. Par conséquent, même un problème simple peut avoir un nombre conséquent de paramètres à optimiser, limitant ainsi le nombre d'algorithmes d'optimisation utilisables en pratique. Depuis 2003, l'algorithme L-BFGS (Limited-memory Broyden Fletcher Goldfarb Shanno), un algorithme quasi-Newton, est le plus utilisé pour l'entraînement d'un CAC [3]. Le lecteur est invité à lire [1] pour plus de détails sur cet algorithme. Les algorithmes à base de gradient stochastique [6] sont également de plus en plus utilisés grâce à leur convergence plus rapide vers une solution similaire à celle obtenue par L-BFGS.

### 2.3 Décodage

Le décodage est le processus qui permet de trouver la meilleure séquence d'étiquettes pouvant être associée à une séquence d'observations donnée. Il consiste à trouver la séquence d'étiquettes  $y^*$  qui maximise la probabilité  $p(y|x)$ . La recherche de la séquence optimale  $y^*$  est réalisée par un algorithme de programmation dynamique de type Viterbi.

$$y^* = \operatorname{argmax}_y(p(y|x))$$

## 3 Définition des fonctions de caractéristiques quantifiées

Dans ce paragraphe, nous posons le problème d'adaptation du formalisme des CAC (basé sur l'utilisation de fonctions de caractéristiques binaires) pour une utilisation avec des observations continues. En effet, contrairement à l'analyse de texte où les observations du système sont des mots, les caractéristiques extraites des images sont généralement des valeurs numériques réelles. Pour pallier<sup>2</sup> cette difficulté, beaucoup de travaux dédiés à l'analyse d'images introduisent une étape de pré-traitement qui consiste à réaliser une classification préalable des données. Les données en sortie de cette étape de classification (des étiquettes, par conséquent des données discrètes) sont ensuite données en entrée d'un modèle CAC. Afin d'exploiter un CAC en tant que système de classification à part entière, nous proposons l'utilisation de fonctions de caractéristiques quantifiées.

Nous définissons une fonction de quantification, notée  $Q()$ , de pas de quantification  $q$  qui quantifie une observation continue  $o$  comme suit :

$$Q(o, q) : \begin{array}{l} 0 \mapsto X \\ o \mapsto x = \operatorname{round}(\frac{o}{q}) \end{array}$$

Si  $o$  est une observation à valeur dans l'intervalle  $[o_{min}, o_{max}]$  alors la fonction de quantification ne peut prendre que  $n$  valeurs discrètes distinctes, avec

$$n = (o_{max} - o_{min})/q$$

Comme nous pouvons le voir, le paramètre de quantification  $q$  ne peut pas être choisi sans avoir de connaissance sur l'étendue des données continues. De plus, toutes les caractéristiques extraites d'une image n'ont pas toujours une distribution uniforme de leurs valeurs. Par exemple, les valeurs les plus petites peuvent être plus discriminantes que les grandes. Dans ce cas, l'utilisation d'une valeur de  $q$  trop grande éliminerait de l'information discriminante; un pas de quantification trop petit diluerait l'information discriminante.

Pour éviter la difficulté du choix d'un pas de quantification, nous proposons d'utiliser un ensemble de fonctions de quantification. Chacune de ces fonctions définit un ensemble de fonctions de caractéristiques binaires. Soit  $q_1, q_2, \dots, q_n$  un ensemble de quantificateurs qui définissent un ensemble de fonctions de quantification  $Q_i(o) = Q(o, q_i)$ , alors, en choisissant une loi dyadique d'évolution des quantificateurs  $q_i = 2 * q_{i-1} = q_1 * 2^{i-1}$ , il est possible de construire un schéma de quantification multi-échelle ayant la capacité de conserver la plupart des informations contenues dans les caractéristiques continues originales sans faire d'hypothèse sur la distribution de ces valeurs. Au final, nous espérons que le CAC sélectionnera les fonctions de caractéristiques quantifiées les plus discriminantes en ajustant leur poids  $\lambda_k$  associé.

Nous pouvons alors redéfinir le modèle général d'un CAC discret en utilisant l'ensemble des  $n$  fonctions de quantification :

$$p(Y|X) = \frac{1}{Z(X)} \exp \left( \sum_t \sum_k \lambda_k f_k(y_{t-1}, y_t, Q_1(o), \dots, Q_n(o)) \right)$$

Le modèle que nous proposons est évalué sur une tâche de segmentation de documents d'archives.

## 4 Expérimentations

Un CAC est un outil performant pour étiqueter des séquences de données en utilisant l'information contextuelle. L'approche CAC que nous proposons est particulièrement adaptée à l'analyse d'images où l'utilisation d'observations continues n'est pas rare. Dans le cadre du projet PlaIR, initié par la région Haute-Normandie, nous souhaitons indexer les archives du "Journal de Rouen", un quotidien paru pendant environ 180 ans. Dans notre expérimentation, nous souhaitons analyser la structure des

pages de journaux afin d'en extraire automatiquement des articles, comme montré dans la figure 2, facilitant ainsi l'indexation de ces archives. Les archives de journaux sont des sources d'information importantes pour les historiens ou pour les particuliers avides de connaissances sur l'histoire de leur région ou de généalogie. Les informations que l'on peut y trouver reflètent l'évolution du contexte social sur une longue période. Une manière efficace de partager cette connaissance est de mettre à disposition ces archives sur Internet, permettant à toute personne de consulter ces journaux sans prendre le risque de dégrader les documents, devenus fragiles avec le temps.

En s'appuyant uniquement sur l'information physique extraite des images de document, notre objectif est de segmenter et d'identifier les lignes de texte, les titres, les séparateurs horizontaux et verticaux, et de différencier ces entités des zones bruitées des images, dues à la numérisation et à la dégradation du papier.

Nous avons choisi de décrire les images de document par des caractéristiques très simples et pertinentes vis-à-vis de la structuration X-Y des pages de journaux : les longueurs de plages de même couleur (Run Length dans la littérature anglophone). Ces caractéristiques sont pertinentes pour des documents fortement structurés horizontalement et sont très simples à utiliser. Chaque pixel est caractérisé par la longueur de la plage horizontale et la longueur de la plage verticale auxquelles il appartient. Les deux valeurs numériques ainsi obtenues pour caractériser un pixel sont des valeurs discrètes mais dont l'étendue des valeurs possibles est grande. La taille moyenne des images est ici de  $1200 \times 1550$ , par conséquent les plages horizontales prennent leur valeur dans l'intervalle  $[1, 1200]$  et les plages verticales dans l'intervalle  $[1, 1550]$ .

Dans un document, la majorité de l'information pertinente est naturellement orientée horizontalement. S'appuyant sur ce constat, nous avons défini un modèle CAC horizontal afin de modéliser des lignes de pixels, ce qui signifie que l'information contextuelle entre les étiquettes ne sera introduite que de manière horizontale. Les caractéristiques horizontales et verticales extraites en chaque pixel sont associées à chaque étiquette locale en utilisant des fonctions de caractéristiques d'ordre 1 (une seule caractéristique est associée à une seule étiquette). L'utilisation d'un tel modèle nous assure l'existence d'un processus de décodage optimal, rapide, mono-dimensionnel similaire à l'algorithme de Viterbi. Une extension bi-dimensionnelle de la méthode nous imposerait l'utilisation d'algorithmes de décodages sous-optimaux comme le loopy belief propagation ou les algorithmes de type coupe de graphe.

Nous présentons maintenant les fonctions de caractéristiques contextuelles que nous utilisons dans cette expérimentation. Nous définissons cinq modèles de combinaison contextuelle du premier ordre sur les caractéristiques horizontales, détaillés dans la figure 1 ( $F_1$  à  $F_5$ ). Ces modèles

rendent compte des dépendances horizontales entre l'étiquette courante et chacune des composantes horizontales définies dans une fenêtre de cinq voisins. Les cinq autres modèles de combinaison contextuels ( $F_6$  à  $F_{10}$ ) rendent compte des dépendances entre l'étiquette courante et chacune des composantes verticales dans une fenêtre horizontale de cinq voisins.

$$\begin{array}{ll} F_1(y_t, o_{t-2}^h) & F_6(y_t, o_{t-2}^v) \\ F_2(y_t, o_{t-1}^h) & F_7(y_t, o_{t-1}^v) \\ F_3(y_t, o_t^h) & F_8(y_t, o_t^v) \\ F_4(y_t, o_{t+1}^h) & F_9(y_t, o_{t+1}^v) \\ F_5(y_t, o_{t+2}^h) & F_{10}(y_t, o_{t+2}^v) \end{array}$$

FIGURE 1 – Modèles de combinaison contextuels des caractéristiques horizontales ( $F_1$  à  $F_5$ ) et des caractéristiques verticales ( $F_6$  à  $F_{10}$ ).

Ces modèles de combinaison définis sur les caractéristiques extraites des images génèrent des fonctions de caractéristiques quantifiées (des Q-fonctions) que nous allons expliciter.

Les petites longueurs de plages apportent de l'information discriminante pour les informations textuelles contenues dans les images, il est donc important de choisir un pas de quantification initial suffisamment petit pour ne pas éliminer cette information discriminante. Dans nos expérimentations nous avons choisi  $q_1 = 2$ . De la même manière, les grandes longueurs de plage ne sont représentatives d'aucune étiquette spécifique. Pour limiter le nombre total de fonctions  $Q()$ , nous limitons donc les valeurs du pas de quantification à l'intervalle  $[2, 512]$ . Nous obtenons donc un processus de quantification multi-échelle à 9 niveaux. Chaque modèle de combinaison  $F_i$  de premier ordre, dont on donne un exemple ci-dessous, génère alors 9 modèles de combinaison des données discrètes quantifiées  $f_i^j$  :

$$\begin{array}{l} f_1^1(y_t, o_{t-2}^h) = F_1(y_t, Q_1(o_{t-2}^h)) \\ f_1^2(y_t, o_{t-2}^h) = F_1(y_t, Q_2(o_{t-2}^h)) \\ \dots \end{array}$$

Ces nouveaux modèles de combinaison génèrent alors les fonctions de caractéristiques binaires comme définies dans la définition originale du L-CAC. Des fonctions binaires de transitions entre étiquettes sont également ajoutées à l'ensemble des fonctions de caractéristiques quantifiées. Au final, le modèle est entraîné en utilisant l'algorithme L-BFGS sur une base d'images étiquetées. L'étape de décodage utilise un algorithme similaire à l'algorithme de Viterbi pour rechercher la séquence d'étiquettes optimale pour chaque ligne de l'image. L'image résultante est reconstruite en concaténant les résultats de chaque ligne de pixels. Pour les expérimentations, nous utilisons la librairie CFR++<sup>1</sup> à laquelle nous avons greffé un module de génération de fonctions de caractéristiques à quantification multi-échelle.

1. <http://crfpp.sourceforge.net/>

## 5 Résultats

Évaluer des résultats de segmentation est un processus difficile, dépendant de la tâche de segmentation. Il est difficile d'évaluer et de quantifier la qualité d'un résultat au niveau du pixel. Aussi, un certain nombre d'articles montrent des images de résultats typiques du comportement de leur méthode pour illustrer les performances du système. Cependant, les compétitions de segmentation doivent quantifier les résultats de manière robuste pour pouvoir effectuer des comparaisons de performances entre méthodes. Certaines tâches de segmentation peuvent se contenter d'un calcul de précision/rappel sur l'étiquetage des pixels pour être évaluées mais certains problèmes de segmentation doivent utiliser des approches pondérées, n'accordant pas la même importance à chaque type d'erreur, pour être correctement évalués. En effet, certaines erreurs dans les résultats de segmentation peuvent être critiques. Par exemple, la segmentation des lignes de texte d'un document est très utile avant l'utilisation d'un OCR mais la performance de cet OCR est très liée à la qualité de la segmentation de la ligne à analyser.

Pour évaluer nos résultats, nous avons décidé d'utiliser le coefficient de similarité de Jaccard (ou indice de Jaccard). Le modèle CAC est entraîné sur une base composée de 11 images, soit 16978 séquences et évalué sur 23 images. Chacune de ces 34 images est complètement étiquetée au niveau pixel par 10 étiquettes distinctes qui caractérisent toutes les entités physiquement présentes dans ces documents. Notons que la combinaison de certaines étiquettes physiques permet d'obtenir un niveau logique d'étiquetage supérieur. Par exemple une ligne de texte est la combinaison des étiquettes caractères, inter-caractères et inter-mots. La liste de ces étiquettes et de leur combinaison est la suivante :

- Séparateur vertical
- Séparateur horizontal
- Titre (composés de "caractères du titre", "inter-caractères du titre" et "inter-mots du titre")
- ligne de texte (composé de "caractères", "inter-caractères" and "inter-mots")
- Bruit
- Fond

L'indice de Jaccard est calculé sur les polygones englobant des étiquettes titre, ligne de texte, séparateurs et zones bruitées.

Pour illustrer l'apport de l'utilisation des fonctions de caractéristiques quantifiées, nous comparons les résultats obtenus avec deux modèles CAC, le premier entraîné avec des Q-fonctions et le second en utilisant des fonctions de caractéristiques classiques (sans quantification mais avec les mêmes fonctions de combinaison) évaluées sur les longueurs de plage extraites des images.

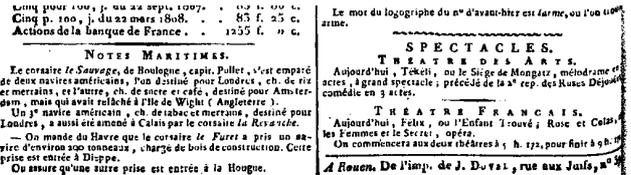


FIGURE 2 – Un exemple de document, avec séparateurs, titres, lignes de texte et zones bruitées

### 5.1 Entraînement du CAC

Le tableau 1 montre le nombre de fonctions de caractéristiques réellement générées à partir des données rencontrées dans la base d'apprentissage, en appliquant les modèles de combinaison présentés dans la section 4, mais aussi le nombre maximum de fonctions de caractéristiques pouvant être générées en théorie (si la base d'apprentissage recouvre tout l'espace des données possibles). Nous observons qu'en pratique, une fraction seulement des fonctions possibles est utilisée (soit 84 760 sur un total possible de 102 000). Les fonctions de caractéristiques dont la réalisation n'apparaît pas dans la base d'apprentissage ne sont pas générées. De plus, 51% des paramètres  $\lambda_k$  ont une valeur (après entraînement) comprise entre  $1.10^{-3}$  et  $-1.10^{-3}$  ce qui signifie que 51% de ces 84,760 fonctions de caractéristiques n'apportent pas de contribution significative par rapport aux valeurs les plus importantes, proches de 4 et -4. Ce nombre de fonctions faiblement pondérées illustre le processus de sélection des fonctions les plus discriminantes opéré pendant l'entraînement du CAC, parmi l'importante quantité de données quantifiées.

Le tableau 1 montre également le temps de calcul nécessaire à l'entraînement des deux systèmes, réalisés sur un processeur cadencé à 2,4 GHz.

TABLE 1 – Nombre de fonctions de caractéristiques et temps d'entraînement

	Sans Q-fonctions	Avec Q-fonctions
Nombre théorique de fonctions	275200	102000
Nombre effectif de fonctions	90520	84850
Temps d'entraînement	9382s	13976s

### 5.2 Résultats de segmentation

La figure 3 montre un exemple de résultat de segmentation en sortie du système.

L'indice de Jaccard utilisé pour évaluer la segmentation des images est calculé suivant cette formule :

$$\frac{VP}{(VP + FP + FN)}$$

où  $VP$  signifie Vrai Positif,  $FP$  signifie Faux Positif et  $FN$  signifie Faux Négatif. Ce coefficient de similarité est calculé sur chaque étiquette évaluée, pour les deux systèmes. Les résultats sont présentés dans le tableau 5.2 et montrent le gain obtenu grâce à l'utilisation de fonctions



FIGURE 3 – Exemple de résultat obtenu en sortie du système

de caractéristiques quantifiées. Ce tableau présente également le nombre de titres et de séparateurs horizontaux correctement identifiés (les deux entités structurelles les plus présentes dans la base d'images) qui augmente également grâce aux Q-fonctions. Notons que même si les séparateurs horizontaux sont quasiment tous détectés sans l'ajout de quantification, la qualité de leur détection est augmentée par l'utilisation des Q-fonctions.

Indices de Jaccard		
Étiquette	Sans Q-fonction	Avec Q-fonctions
Sep H	0.8223	0.8919 (+0.07)
Sep V	0.9136	0.9641 (+0.05)
Titres	0.7503	0.8317 (+0.08)
Lignes de texte	0.9789	0.986 (+0.007)
Zones bruitées	0.2876	0.4078 (+0.12)
Nombre d'entités		
Sep H	204/212(96.23%)	211/212(99.53%)
Titres	106/127(83.46%)	120/127(94.49%)

TABLE 2 – Indices de Jaccard calculés sur les séparateurs horizontaux, les séparateurs verticaux, les titres, les lignes de texte et les zones de bruit pour les deux modèles CAC, ainsi que le nombre de titres et de séparateurs horizontaux correctement détectés

Comme nous l'avons signalé précédemment, une petite er-

reur de segmentation peut avoir un impact important sur les étapes de post-traitement. Pour notre tâche, manquer un séparateur horizontal signifie manquer le début d'un nouvel article. Mais l'erreur la plus critique survient quand un séparateur vertical est étiqueté comme caractère. Cette petite erreur signifie que les lignes de textes de chaque coté du séparateur vertical (et donc appartenant à deux colonnes de texte distinctes) ont été concaténées ce qui rompt le sens de lecture du document. Un bon indicateur de ce type d'erreur est le taux de confusion entre l'étiquette "caractère" et l'étiquette "séparateur vertical". L'utilisation des Q-fonctions permet de diminuer ce taux de confusion, passant alors de 3,15% à 1,47%, réduisant ainsi le nombre d'erreurs critiques pour la segmentation des lignes de texte.

Le temps de décodage d'une image est relativement faible et dépend de la taille de l'image mais aussi du nombre de fonctions de caractéristiques utilisées dans le modèle. Pour 84850 fonctions de caractéristiques, le modèle décode une image en 2,7 secondes en moyenne. Pour un million de fonctions de caractéristiques le temps de décodage est doublé.

### 5.3 Comparaison avec un système CAC continu

Jusqu'à maintenant, nous utilisons un modèle CAC discret, basé sur des fonctions de caractéristiques binaires. Il est cependant possible d'utiliser un modèle CAC continu, dont les fonctions de caractéristiques ne sont pas binaires. Cette partie de l'article compare les résultats obtenus avec un CAC continu et les résultats précédents. Pour faire cette expérimentation, nous avons utilisé la librairie HCRF<sup>2</sup> configurée en mode "crf". Les données d'entrée restent des longueurs de plages de couleur identique et les fonctions de caractéristiques sont alors ces données observées. La librairie utilisée permet de définir la taille d'une fenêtre de combinaison des observations, noté  $W$ . Nous avons fait varier ce paramètre de 0 à 2. Nous utilisons également un terme de type régularisation L2 (de paramètre 0.25 défini expérimentalement) lors de l'apprentissage du modèle. Les performances sont évaluées avec une stratégie identique à celle utilisée pour l'obtention des résultats précédents, les valeurs obtenus sont donc parfaitement comparables entre elles.

Indices de Jaccard pour CAC continu			
Étiquette	$W = 0$	$W = 1$	$W = 2$
Sep H	0.5462 (-0.35)	0.7059 (-0.19)	0.4049 (-0.49)
Sep V	0.8225 (-0.14)	0.7822 (-0.18)	0.6859 (-0.28)
Titres	0.3635 (-0.47)	0 (-0.83)	0.0011 (-0.83)
Texte	0.9348 (-0.05)	0.8690 (-0.12)	0.8756 (-0.11)
Bruit	0.0319 (-0.38)	0 (-0.41)	0.1526 (-0.26)

TABLE 3 – Indices de Jaccard obtenus par le CAC continu pour 3 tailles de fenêtre de combinaison, relativement au CAC discret avec Q-fonctions.

2. <http://sourceforge.net/projects/hcrf/>

Les résultats sont donnés dans la table 5.3 et montrent qu'un CAC continu, dans notre expérimentation, ne parvient pas à modéliser les données d'apprentissage aussi bien que le système présenté précédemment. Les lignes de texte sont relativement bien modélisées mais les étiquettes plus complexes tels que l'étiquette titre, n'ont quasiment pas été modélisées par le CAC continu.

## 6 Conclusions

Dans cet article nous avons proposé d'utiliser des fonctions de quantification de caractéristiques continues multi-échelles dans un modèle de champs aléatoire conditionnel. Le modèle CAC est utilisé comme un modèle de ligne de pixels et évalué sur une tâche de segmentation d'images de documents numérisés.

Les résultats obtenus montrent que l'utilisation de ce type de fonctions de caractéristiques permet d'obtenir une meilleure segmentation sur des journaux anciens et d'étiqueter des entités dans le document. Nous avons comparé nos résultats avec ceux obtenus avec un CAC continu sur cette même tâche et nous pouvons en conclure que pour notre expérimentation, les meilleurs résultats sont obtenus grâce à notre approche multi-échelle. Le système présenté peut être facilement adapté à d'autres tâches de segmentation en utilisant d'autres caractéristiques physiques extraites des images.

Ces résultats montrent que l'utilisation de modèles linéaires permet d'obtenir une segmentation de données bi-dimensionnelles en utilisant un système efficace, dont la complexité est maîtrisée. L'utilisation d'un modèle de séquence 1D permet également l'utilisation d'outils de programmation disponibles sur Internet.

Pour compléter ces résultats, nous envisageons d'étudier l'influence d'une sélection des fonctions de caractéristiques quantifiées après apprentissage des paramètres du CAC. Le fait d'avoir une hiérarchie de quantification signifie que l'information apportée par un ensemble de valeurs quantifiées peut d'une certaine manière être résumée par une valeur de niveau hiérarchique supérieur. Cette sélection permettrait alors une diminution du nombre de paramètres du système (et du temps de décodage) et une généralisation de la connaissance acquise par le système.

## Références

- [1] J. Nocedal and S. J. Wright. *Large scale quasi-Newton and partially separable optimization*, in "Numerical Optimization", Chapter 9. Springer, 1999.
- [2] J. Lafferty, A. McCallum and F. Pereira. *Conditional random fields : probabilistic models for segmenting and labeling sequence data*, Proc. 18th International Conf. on Machine Learning. San Francisco, 2001.
- [3] F. Sha and F. Pereira. *Shallow parsing with conditional random fields*, Proc. NAACL '03. Stroudsburg, USA, 2003.
- [4] T. M. Breuel. *High performance document layout analysis*, in Symposium on Document Image Understanding Technology, Greenbelt. USA, 2003.
- [5] X. He, R. S. Zemel and M. A. Carreira-Perpinan. *Multiscale conditional random fields for image labeling*, Proc. In CVPR. Washington DC, USA, 2004.
- [6] S. Vishwanathan, N. N. Schraudolph, M. W. Schmidt and K. Murphy. *Accelerated Training of Conditional Random Fields with Stochastic Gradient Methods*, Proc. ICML '06. , New-York, USA, 2006.
- [7] C. Sutton and A. McCallum. *Introduction to conditional random fields for relational learning*, In "Introduction to statistical relational learning", chapter 1. 2006.
- [8] T. M. T. Do and T. Artières. *Conditional random fields for online handwriting recognition*, IWFHR. La Baule, France, 2006.
- [9] S. Feng, R. Manmatha, and A. McCallum. *Exploring the use of conditional random field models and HMMs for historical handwritten document recognition*, DIAL. Washington, DC, USA, 2006.
- [10] S. Nicolas, J. Dardenne, T. Paquet and L. Heutte. *Document image segmentation using a 2D conditional random field model*, Proc. ICDAR '07. Curitiba, Brazil, 2007.
- [11] C.-H. Lee, S. Wang, A. Murtha, M. R. G. Brown and R. Greiner. *Segmenting brain tumors using pseudo-conditional random fields*, Proc. MICCAI '08. New York City, USA, 2008.
- [12] S. Chaudhury, M. Jindal and S. Dutta Roy. *Model-guided segmentation and layout labelling of document images using a hierarchical conditional random field*, Proc. PReMI '09. Berlin, Germany, 2009.