REVUE DE STATISTIQUE APPLIQUÉE

T. BROUARD
M. SLIMANE
J.-P. ASSELIN DE BEAUVILLE
G. VENTURINI

Apprentissage d'une chaîne de Markov cachée. Problèmes numériques liés à l'application à l'image

Revue de statistique appliquée, tome 46, n° 2 (1998), p. 83-108
<http://www.numdam.org/item?id=RSA_1998__46_2_83_0>

Rev. Statistique Appliquée, 1998, **XLVI** (2), 83-108

APPRENTISSAGE D'UNE CHAÎNE DE MARKOV CACHÉE. PROBLÈMES NUMÉRIQUES LIÉS À L'APPLICATION À L'IMAGE

T. Brouard, M. Slimane, J.-P. Asselin de Beauville, G. Venturini

Laboratoire d'Informatique, Ecole d'Ingénieurs en Informatique pour l'Industrie
Université de Tours, 64, Avenue Jean Portalis, 37200 Tours (France)

RÉSUMÉ

Cet article traite de la résolution des problèmes numériques liés à l'apprentissage des chaînes de Markov cachées. Ces problèmes ayant une forte probabilité d'apparition lors de l'emploi de ces modèles pour l'analyse des images numériques nous avons choisi d'illustrer nos approches sur ce champ d'application. Les solutions que nous proposons sont des extensions des méthodes de rescaling basées sur un seul vecteur d'observations au cas de plusieurs vecteurs d'observations. Les tests que nous avons réalisés justifient pleinement ces solutions par une diminution très importante des valeurs de probabilité voisines de zéro.

Mots-clés : *Chaînes de Markov cachées, apprentissage d'images, underflow*

ABSTRACT

This paper deals with solving numerical problems that appear when learning hidden Markov models. These problems have a high probability of appearing when dealing with image processing, and we have therefore used this domain to illustrate our approach. The solution that we propose consists in extending rescaling technique for single vector of observations to multiple vectors. The tests that we have performed justify this solution and show that less low probability values are obtained.

Keywords : *Hidden Markov models, picture learning, underflow*

1. Introduction

Les chaînes de Markov sont des outils de modélisation très utilisés dans de nombreux domaines, en reconnaissance des formes plus particulièrement. Le modèle standard peut être étendu aux «chaînes de Markov cachées» (CMC) pour lesquelles on peut distinguer deux composantes principales : les observations, qui sont observables et les états cachés inobservables. A chaque état caché d'une CMC est associé une distribution de probabilités d'émission de symboles observables. A partir d'une suite observée de tels symboles il est possible d'apprendre (ou d'estimer) les paramètres

d'une CMC susceptible de produire cette suite avec une forte probabilité. Le nombre de «degrés de liberté» (paramètres) important de ce type de processus stochastique est un facteur prépondérant pour justifier l'emploi des CMC pour la modélisation de processus stochastiques complexes tels que : la reconnaissance de la parole [15], la modélisation du langage [14], la reconnaissance de caractères manuscrits [3], le traitement du signal [20] ou l'analyse des images [12][17][21]. Il existe de nombreux algorithmes d'apprentissage des paramètres d'une CMC qui sont basés, en général, sur des méthodes de gradient tel, par exemple, que l'algorithme de Baum-Welch. Un certain nombre de travaux récents visent à améliorer les performances de ces méthodes itératives en faisant appel à une exploration parallèle de l'espace des paramètres [25][4]. La mise en œuvre, dans tous les cas, de ces algorithmes d'apprentissage est souvent délicate et difficile à cause des nombreux problèmes numériques rencontrés. En effet, ces méthodes reposent sur des produits de probabilités qui ont de grandes chances de s'annuler dès que le nombre de facteurs du produit augmente. Dans cet article on propose une méthode de «rescaling» dont l'objectif est précisément d'éviter ces problèmes lorsque le nombre de facteurs en cause devient grand ou très grand. Afin d'illustrer le comportement de ces algorithmes on les applique à l'apprentissage d'images numériques, problème qui, par nature, génère des produits de probabilités de grande taille. La suite de cet article est organisée comme suit : le paragraphe 2 rappelle les principales définitions et les problèmes à résoudre pour mettre en œuvre une CMC. Les paragraphes 3 et 4 présentent les problèmes numériques rencontrés et les solutions qui ont été proposées dans le passé. Dans le paragraphe 5 on propose une adaptation des CMC et des algorithmes d'apprentissage traditionnels au contexte particulier de l'image numérique. Le paragraphe 6 présente notre approche des problèmes numériques de l'apprentissage d'une CMC appliqué à l'image. Le paragraphe 7 contient des résultats numériques comparant nos méthodes de «rescaling vectoriel» avec les approches classiques pour quelques images. Enfin la section 8 contient une brève conclusion.

2. Chaînes de Markov cachées (CMC) : définitions

2.1. Définitions formelles

Une chaîne de Markov cachée est une chaîne de Markov stationnaire générant un processus stochastique à deux composantes : une cachée, l'autre observable. La composante cachée est la réalisation $q = (q_1, q_2, \ldots, q_T)$ d'une suite de variables aléatoires $Q = (Q_1, Q_2, \ldots Q_T)$ où les q_i prennent leurs valeurs dans l'ensemble S de N classes (les états). La suite observable $o = (o_1, o_2, \ldots, o_T)$ est la réalisation du processus aléatoire $(O_1, O_2, \ldots, O_T)$ où les o_i prennent leurs valeurs dans l'ensemble V de M classes (les symboles). La propriété de Markov s'applique sur la composante cachée :

$$\forall s \in N, \quad P(Q_s = q_s | Q_1 = q_1 \ldots Q_{s-1} = q_{s-1}) = P(Q_s = q_s | Q_{s-1} = q_{s-1})$$

Et l'on a pour cette composante les propriétés suivantes :

- loi initiale : $P(Q_1 = q_i) = \pi_i \quad 1 \leqslant i \leqslant N$
- matrice de transition : $P(Q_s = q_j | Q_{s-1} = q_i) = a_{ij} \quad 1 \leqslant i, j \leqslant N$

Les éléments d'une CMC sont donc :

• N : le nombre d'états du modèle. Les états forment l'ensemble $S = \{s_1, s_2, \ldots, s_N\}$.

• M : le nombre de symboles différents générés par la CMC. Ces symboles forment l'ensemble $V = \{v_1, v_2, \ldots v_M\}$.

• Matrice de distribution des probabilités de transition : $A = \{a_{ij}\}$. Cette matrice est de dimension $N \times N$. Le terme générique a_{ij} désigne la probabilité de transiter d'un état s_i vers un état s_j. La matrice est stochastique, *i.e.* la somme de tous les a_{ij} d'une ligne de la matrice est égale à 1.

• Matrice de distribution des probabilités de génération des symboles : $B = b_j(v)$. Cette matrice est de dimension $N \times M$. Le terme générique $b_j(v)$ désigne la probabilité pour que la CMC se trouve dans l'état s_j et génère le symbole v. La matrice B est stochastique.

• Matrice de distribution des probabilités initiales : $\Pi = \{\pi_i\}$. Cette matrice est de dimension $1 \times N$. Le terme générique π_i désigne la probabilité que la CMC se situe dans l'état s_i à l'instant initial. La matrice Π est stochastique.

En résumé, la CMC est définie par le triplet de matrices noté $\lambda = (A, B, \Pi)$.

2.2. Exemple

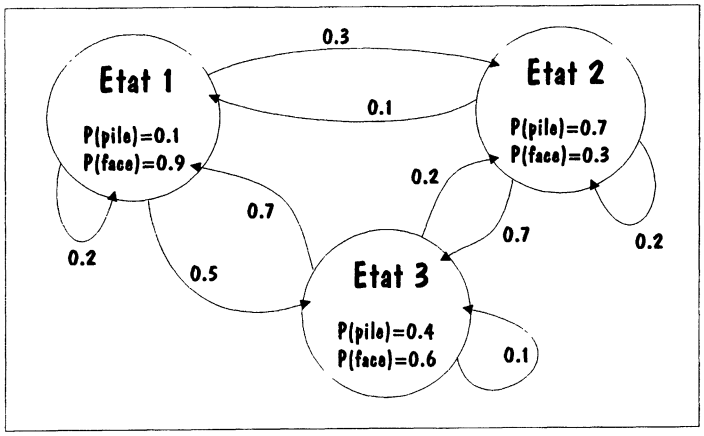
Considérons un lancer de pièce de monnaie. Nous disposons de trois pièces (biaisées). Le lanceur ne nous informe pas de la pièce qu'il a choisie, et pour une même observation donnée, les symboles (pile ou face) peuvent être générés tour à tour par la pièce une deux et trois. Une *chaîne de Markov cachée* modélise l'expérience de la manière suivante. On considère qu'un état modélise une pièce, et que cet état peut produire deux symboles : pile et face. On admet alors que l'on peut passer d'un état à un autre (c'est-à-dire changer de pièce) suivant certaines lois de probabilités, et que chaque pièce possède sa propre distribution de probabilités pour générer les symboles pile et face. La séquence de (pile-face) est toujours observable (matrice B), mais la séquence d'états qui a engendré cette séquence n'est pas observable (matrice A). On dit alors qu'elle est cachée.

2.3. Problèmes fondamentaux liés aux CMC

Trois types de problèmes se posent en relation avec l'emploi d'une CMC :

2.3.1. Problème d'évaluation

Evaluation de la probabilité de générer une observation : étant donnés une observation notée o de longueur T et une CMC notée λ, il s'agit d'évaluer la probabilité avec laquelle la CMC λ peut engendrer l'observation o notée $P(o|\lambda)$. Cette valeur peut bien sûr être calculée directement, mais le problème est résolu par l'algorithme Forward qui permet le calcul avec une complexité bien inférieure (N^2T opérations) au lieu de $2TN^T$ opérations pour le calcul direct.



$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.1 & 0.2 & 0.7 \\ 0.7 & 0.2 & 0.1 \end{bmatrix} \end{array} \qquad B = \begin{array}{c} \begin{array}{cc} P & F \end{array} \\ \begin{bmatrix} 0.1 & 0.9 \\ 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \end{array} \qquad \Pi = \begin{bmatrix} 0.1 & 0.5 & 0.4 \end{bmatrix}$$

Exemple d'observation : (avec P=pile et F=face) : o = PPPFFPFFPPPFPP

FIGURE 1
Exemple de chaîne de Markov cachée : modélisation d'un lancer avec 3 pièces de monnaie

2.3.2. Problème du meilleur chemin d'états

Trouver le chemin d'états optimal : étant donné une observation notée o et une CMC notée λ, déterminer le chemin d'états noté q le plus probablement suivi lors de la génération de l'observation o par la CMC λ. Ce qui se note $P(q|o, \lambda)$ maximal. Ce chemin est déterminé par l'algorithme de Viterbi, qui fait appel à des techniques de programmation dynamique.

2.3.3. Problème d'optimisation supervisée d'une CMC

Optimisation supervisée du modèle : étant donné une observation notée o, déterminer le modèle λ d'architecture connue tel que sa probabilité de générer l'observation o soit maximale. Plusieurs algorithmes existent dans ce domaine dont le «k-segmental means» [18] et l'algorithme de Baum-Welch. Le premier réestime les valeurs du modèle en faisant un simple comptage en association avec l'algorithme de Viterbi, alors que le second utilise un gradient.

2.3.4. Problème d'optimisation non-supervisée d'une CMC

Optimisation non-supervisée du modèle : étant donné une observation notée o, déterminer le modèle λ d'architecture inconnue tel que sa probabilité de générer l'observation o soit maximale. Nous proposons dans ce domaine un algorithme génétique hybride appelé GHOSP utilisant l'algorithme de Baum-Welch [10].

3. Problèmes numériques liés à l'optimisation

L'optimisation d'un modèle de Markov caché par l'algorithme de Baum-Welch nécessite le calcul de variables notées $\alpha_t(i)$ et $\beta_t(i)$. Ces variables sont définies comme suit :

$$\alpha_t(i) = P(o_1 o_2 \dots o_{t-1} o_t; Q_t = s_i|\lambda) \quad t \in 1 \dots T$$
$$\beta_t(i) = P(o_{t+1} \dots o_T, Q_t = s_i|\lambda) \quad t \in 1 \dots T$$

Ces variables sont interprétées de la manière suivante :

• $\alpha_t(i)$ est la probabilité que la chaîne de Markov cachée λ génère l'observation partielle $o_1 o_2 \dots o_{t-1} o_t$ et que cette chaîne soit dans l'état s_i à l'instant t (Q_t est la variable aléatoire modélisant l'état caché de la chaîne à l'instant t).

• $\beta_t(i)$ est la probabilité que la chaîne de Markov cachée λ génère l'observation partielle $o_{t+1} \dots o_T$ et que cette chaîne soit dans l'état s_i à l'instant t.

Le calcul de $\alpha_t(i)$ et $\beta_t(i)$ est réalisé récursivement, sous forme de produits de probabilités. Ces produits tendent d'autant plus vite vers zéro que les termes qui les composent sont nombreux. Le nombre de ces termes est directement lié à la longueur de l'observation que l'on veut faire apprendre à la CMC.

Ce phénomène est appelé *underflow*. Il se traduit par le fait qu'au delà d'un certain point les nombres deviennent trop petits pour être manipulés par la machine. Le calculateur considère que ces nombres sont nuls et entraîne des imprécisions, voire des erreurs de calcul lors de divisions par exemple.

Dans l'état actuel, une CMC ne peut pas apprendre une observation «trop longue». Dans la pratique, on peut estimer la longueur limite à quelques dizaines de symboles, suivant les problèmes. Le problème est encore plus difficile dans le cas de l'apprentissage d'images, présentant une grande quantité de données.

4. Solutions classiques au problème de l'underflow

Classiquement on résout le problème de l'underflow dans l'algorithme de Baum-Welch par des méthodes (dites de «rescaling») introduisant des facteurs de correction (appelés facteurs de rescaling). Le cas typique ([18][13]) consiste à normaliser les valeurs des $\alpha_t(i)$ et $\beta_t(i)$. On corrige ensuite l'effet de cette normalisation lors de la réestimation des paramètres de la CMC. L'inconvénient majeur est l'augmentation significative du coût en nombre d'opérations des algorithmes. Une seconde méthode [12] consiste à remplacer les probabilité conjointes ($P(o_1 o_2 \dots o_t, Q_t = s_i|\lambda)$) par

les probabilités *a posteriori* ($P(Q_t = s_i|o_1 o_2 \ldots o_t, \lambda)$) dans les calculs des variables Forward ($\alpha_t(i)$) et Backward ($\beta_t(i)$). Nous ne nous intéresserons ici qu'aux seules méthodes de rescaling car la seconde approche n'est qu'une approximation de la théorie.

4.1. Rappel des relations de base [18]

4.1.1. Variable forward

$$\alpha_t(j) = P(o_1 o_2 \ldots o_t, Q_t = s_j|\lambda) = \sum_{i=1}^{N} (\alpha_{t-1}(i) a_{ij}) \cdot b_j(o_t) \begin{cases} \forall t \in 1 \ldots T-1 \\ \forall j \in 1 \ldots N \end{cases} \tag{1}$$

Ces coefficients sont regroupées dans la matrice α. Une ligne d'indice t est associée au symbole de rang t dans l'observation. Une colonne d'indice j est associée à l'état caché s_j. A l'intersection de cette ligne et de cette colonne, on trouve le coefficient $\alpha_t(j)$.

4.1.2. Variable backward

$$\beta_t(i) = P(o_{t+1} \ldots o_{T-1} o_T, Q_t = s_i|\lambda) = \sum_{j=1}^{N} (a_{ij} \beta_{t+1}(j) b_j(o_t)) \begin{cases} \forall t \in 1 \ldots T-1 \\ \forall i \in 1 \ldots N \end{cases} \tag{2}$$

Ces coefficients sont regroupées dans la matrice β. Une ligne d'indice t est associée au symbole de rang t dans l'observation. Une colonne d'indice i est associée à l'état caché s_i. A l'intersection de cette ligne et de cette colonne, on trouve le coefficient $\beta_t(i)$.

4.1.3. Calcul de $P(o|\lambda)$

$$P(o|\lambda) = \sum_{i=1}^{N} \alpha_T(i) = \sum_{i=1}^{N} \alpha_t(i) \beta_t(i) \quad \forall t \in 1 \ldots T \tag{3}$$

4.2. Rescaling de la matrice α

La méthode proposée par Rabiner [18] consiste à calculer une ligne de la matrice α puis à normaliser celle-ci de telle sorte que la somme des éléments de cette ligne soit égale à 1. Pour chaque ligne d'indice t de cette matrice on introduit donc un coefficient de normalisation c_t définit par :

$$c_t = \frac{1}{\sum_{i=1}^{N} \alpha_t(i)} \tag{4}$$

A la première itération, on calcule $\alpha_1(i) = \pi_i \cdot b_i(o_1) \quad \forall i \in 1 \dots N$
On peut alors calculer le coefficient de normalisation de la première ligne c_1 :

$$c_1 = \frac{1}{\sum_{i=1}^{N} \alpha_1(i)} \tag{5}$$

On normalise alors la première ligne de α pour obtenir les coefficients $\tilde{\alpha}_1(i)$:

$$\tilde{\alpha}_1(i) = \alpha_1(i) \cdot c_1 \quad \forall i \in 1 \dots N \tag{6}$$

A la seconde itération, (sans rescaling) on a :

$$\alpha_2(j) = \sum_{i=1}^{N} (\alpha_1(i) \cdot a_{ij}) \cdot b_j(o_2) \quad \forall i \in 1 \dots N \tag{7}$$

Posons :

$$\alpha'_2(j) = \sum_{i=1}^{N} (\tilde{\alpha}_1(i) \cdot a_{ij}) \cdot b_j(o_2) \quad \forall i \in 1 \dots N \tag{8}$$

ce qui donne :

$$\begin{aligned} \alpha'_2(j) &= \sum_{i=1}^{N} (c_1 \cdot \alpha_1(i) \cdot a_{ij}) \cdot b_j(o_2) \\ &= c_1 \cdot \sum_{i=1}^{N} (\alpha_1(i) \cdot a_{ij}) \cdot b_j(o_2) \quad \forall i \in 1 \dots N \end{aligned} \tag{9}$$

$$\Leftrightarrow \quad \alpha'_2(j) = c_1 \cdot \alpha_2(j) \qquad \forall i \in 1 \dots N \tag{10}$$

Puis on normalise ces valeurs :

$$\tilde{\alpha}_2(j) = c_2 \cdot \alpha'_2(j) = c_1 \cdot c_2 \cdot \alpha_2(j) \quad \forall i \in 1 \dots N \tag{11}$$

Pour le terme général $\tilde{\alpha}_{t+1}(j)$ on a :

$$\tilde{\alpha}_{t+1}(j) = \left(\prod_{s=1}^{t+1} c_s \right) \cdot \alpha_{t+1}(j) \quad \forall i \in 1 \dots N \tag{12}$$

NB. Dans la suite de l'exposé, on note : $C_t = \prod_{s=1}^{t} c_s$

4.3. Modification du calcul de $P(o|\lambda)$

Soit :

$$P(o|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{13}$$

En utilisant le rescaling, on a :

$$\sum_{i=1}^{N} \tilde{\alpha}_T(i) = \sum_{i=1}^{N} C_T \cdot \alpha_T(i) = C_T \cdot \sum_{i=1}^{N} \alpha_T(i) = C_T \cdot P(O|\lambda) \tag{14}$$

D'où :

$$P(o|\lambda) = \frac{1}{C_T} \cdot \sum_{i=1}^{N} \tilde{\alpha}_T(i) \tag{15}$$

D'après la définition du rescaling, on a : $\begin{cases} C_T = \dfrac{1}{\sum_{i=1}^{N} \alpha_T(i)} \\ \tilde{\alpha}_T(i) = C_T \cdot \alpha_T(i) \end{cases}$

D'où :

$$\sum_{i=1}^{N} \tilde{\alpha}_T(i) = \sum_{i=1}^{N} C_T \cdot \alpha_T(i) = \frac{1}{\sum_{i=1}^{N} \alpha_T(i)} \cdot \sum_{i=1}^{N} \alpha_T(i) = 1 \tag{16}$$

Donc l'expression de $P(o|\lambda)$ est égale à $\frac{1}{C_T}$.

4.4. Rescaling de la matrice β

L'ordre de grandeur des $\beta_t(i)$ étant sensiblement le même que celui des $\alpha_t(i)$, les mêmes coefficients de rescaling leur sont appliqués. Cela permet également de n'induire pratiquement aucune modification dans les formules de réestimation des $a_{ij}, b_j(o_t)$ et π_i.

La démonstration du calcul de $\tilde{\beta}_t(i)$ est sensiblement la même que celle de $\tilde{\alpha}_t(i)$.

Sans rescaling , on a :

$$\beta_T(i) = 1 \qquad \forall i \in 1 \ldots N \tag{17}$$

$$\beta_{T-1}(i) = \sum_{j=1}^{N} (a_{ij} \cdot b_j(o_T) \cdot \beta_T(j)) \quad \forall i \in 1 \ldots N \tag{18}$$

Ce qui devient avec rescaling :

$$\tilde{\beta}_T(i) = c_T \cdot \beta_T(i) \quad \forall i \in 1 \dots N \tag{19}$$

Posons :

$$\beta'_{T-1}(i) = \sum_{j=1}^{N} (a_{ij} \cdot b_j(o_T) \cdot \tilde{\beta}_T(j)) \quad \forall i \in 1 \dots N \tag{20}$$

D'où, en remplaçant $\tilde{\beta}_T(i)$:

$$\beta'_{T-1}(i) = c_T \cdot \sum_{j=1}^{N} (a_{ij} \cdot b_j(o_T) \cdot \beta_T(j)) = c_T \cdot \beta_{T-1}(i) \quad \forall i \in 1 \dots N \tag{21}$$

La normalisation de ces coefficients donne alors :

$$\tilde{\beta}_{T-1}(i) = c_{T-1} \cdot \beta'_{T-1}(i) = c_{T-1} \cdot c_T \cdot \beta_{T-1}(i) \quad \forall i \in 1 \dots N \tag{22}$$

Le terme général $\tilde{\beta}_{t+1}(i)$ est donné par :

$$\tilde{\beta}_{t+1}(i) = \left(\prod_{s=t+1}^{T} c_s \right) \cdot \beta_{t+1}(i) \quad \forall i \in 1 \dots N \tag{23}$$

NB. Dans la suite de l'exposé, on note : $D_{t+1} = \prod_{s=t+1}^{T} c_s$

- les valeurs sans rescaling sont notées $\alpha_t(i), \beta_t(i), a_{ij}, b_j(o_t), \pi_i$
- les valeurs avec rescaling sont notées $\tilde{\alpha}_t(i), \tilde{\beta}_t(i)$
- les réestimations des paramètres sans rescaling de la CMC sont notées $\overline{a}_{ij}, \overline{b}_j(o_t), \overline{\pi}_i$
- les réestimations des paramètres avec rescaling de la CMC sont notées ${}^r\overline{a}_{ij}, {}^r\overline{b}_j(o_t), {}^r\overline{\pi}_i$

4.5. Formules de réestimation des paramètres d'une CMC dans le cas d'un seul vecteur d'observations

$$ {}^{r}\overline{\pi}_i = \frac{\tilde{\alpha}_1(i) \cdot \tilde{\beta}_1(i)}{c_1} \quad \forall i \in 1 \ldots N \tag{24} $$

$$ {}^{r}\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) \cdot a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \sum_{h=1}^{N} \alpha_t(i) \cdot a_{ih} \cdot b_h(o_{t+1}) \cdot \beta_{t+1}(h)} \quad \begin{matrix} \forall i \in 1 \ldots N \\ \forall j \in 1 \ldots N \end{matrix} \tag{25} $$

$$ {}^{r}\overline{b}_j(s) = \frac{\sum_{t=1 \wedge o_t = v_s}^{T} \frac{\tilde{\alpha}_t(j) \cdot \tilde{\beta}_t(j)}{c_t}}{\sum_{t=1}^{T} \frac{\tilde{\alpha}_t(j) \cdot \tilde{\beta}_t(j)}{c_t}} \quad \begin{matrix} \forall j \in 1 \ldots N \\ \forall s \in 1 \ldots M \end{matrix} \tag{26} $$

Dans la formule (26), la sommation n'est effectuée que pour les indices $t \in \{1, \ldots, T\}$ pour lesquels $o_t = v_s$ d'ou la notation $t = 1 \wedge (o_t = v_s)$ pour la valeur initiale de l'indice de la somme. Ces relations sont difficilement applicables lorsque la longueur d'une observation devient très grande (de l'ordre de quelques milliers à quelques dizaines de milliers). La raison est liée à la correction du calcul de $P(o|\lambda)$. En effet, la probabilité d'une observation est :

$$ P(o|\lambda) = \frac{1}{\prod_{t=1}^{T} c_t} \tag{27} $$

où c_t désigne le coefficient de normalisation pour le symbole de rang t dans l'observation. Les probabilités étant petites, les coefficients de normalisation c_t sont grands. Lorsque T est très grand, le produit des coefficients c_t devient tellement grand que la précision en machine est insuffisante pour le représenter. Le calculateur assimile alors ce nombre à l'infini. Un nouveau phénomène apparaît : *l'overflow*. Nous proposons ci-dessous une solution nous permettant la résolution de l'underflow en évitant celui de l'overflow.

5. Solution vectorielle du problème de l'underflow – Application à l'image numérique

L'apprentissage d'une image (en apprenant les niveaux de gris des pixels) produit des observations de très grande taille ce qui implique, comme on l'a vu ci-dessus des problèmes de calcul.

Deux orientations sont alors possibles :

1) coder l'image différemment (par exemple en la découpant en zones et en codant la bande passante de chaque zone [11]).

2) poser le problème différemment, de façon plus adaptée à l'image et introduire de nouveaux outils.

Nous allons présenter la seconde approche. Nous introduirons en premier lieu une architecture de CMC mieux adaptée à la modélisation de l'image. Puis nous décrirons une méthode d'apprentissage en correspondance avec la nouvelle architecture. Cette méthode permet de contourner de façon efficace les problèmes de l'underflow et d'overflow, avec un coût de calcul bien moindre.

5.1. Les modèles de Markov cachés pseudo-2D

Ces modèles ont été appelés modèles de Markov cachés pseudo-2D (CMC p2D) par Agazzi *et al.* [1] dans leurs travaux sur la reconnaissance de mots-clés. La dénomination «pseudo-2D» vient de leur architecture dont les états ne sont pas complètement connectés entre eux. Leur emploi se justifie par le fait que ces modèles sont mieux adaptés à la modélisation d'images [23][26] comparé aux chaînes de Markov cachées ergodiques ou gauche-droite [18]. Nous utilisons donc ce type de modèle afin de déterminer l'architecture de la CMC p2D la mieux adaptée à l'image que l'on désire apprendre, sachant que l'on détermine par la suite une architecture de CMC équivalente (*cf.* 5.1.2 Passage d'une CMC p2D vers une CMC).

5.1.1. Présentation des CMC p2D

Une CMC p2D est une chaîne de Markov (CM) composée de «super-états». Le fonctionnement de cette CM est régi par deux matrices : A et Π. Chaque super-état représente une CMC. Globalement, chaque CMC est associée à une région de l'image, et la CMC p2D gère la transition d'une CMC à l'autre, permettant ainsi la constitution de l'image en juxtaposant les différentes régions modélisées par les CMC. Pour cette raison, les CMC p2D sont des modèles associés à un sens de balayage de l'image (généralement de la gauche vers la droite et de haut en bas) [18].

L'architecture des modèles pseudo-2D peut se déduire assez facilement de la structure de l'image. La figure 2 montre une image simple et une architecture de CMC pseudo-2D possible. L'image est découpée en 5 bandes verticales à chacune desquelles on associe un super-état. Chaque bande verticale est ensuite parcourue de haut en bas en associant à chaque transition de niveau de gris un état caché d'une CMC.

5.1.2. Passage d'une CMC p2D vers une CMC

L'architecture plus complexe des CMC p2D nécessite l'introduction de nouveaux algorithmes d'évaluation et d'apprentissage. Pour contourner ce problème, on utilise des algorithmes de passage d'une CMC p2D vers une CMC équivalente. On obtient alors une CMC, ce qui permet la réutilisation des algorithmes des CMC. Deux méthodes de passage d'une CMC p2D vers une CMC (avec et sans introduction

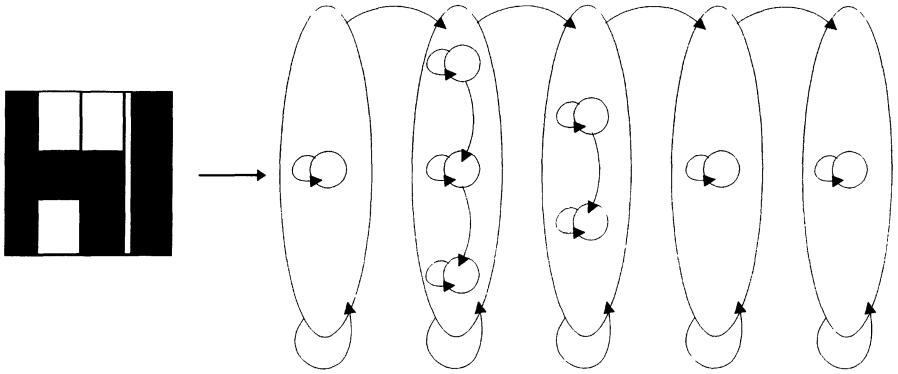


FIGURE 2
Exemple d'image («hl») et d'architecture de CMC pseudo-2D associée

d'états cachés fictifs) sont décrites dans [22] et [8]. A titre d'exemple, nous donnons ici le principe de celle que nous utilisons.

L'idée de base est d'éliminer la notion de super-état en reportant les transitions s'y rattachant sur les états de début et de fin des CMC concernées. Par exemple, la transition du super-état 1 (constitué de la CMC 1) vers le super-état 2 (constitué de la CMC 2) est remplacée par une transition du dernier état caché de la CMC 1 vers le premier état caché de la CMC 2. De même, la probabilité de rester dans le même super-état se traduit par un probabilité de passer du dernier état caché vers le premier état caché de la même CMC. Les probabilités associées à ces transitions sont réévaluées par l'algorithme de Baum-Welch, et ne posent donc aucun problème de calcul. La figure 3 donne un exemple de transformation d'une CMC p2D en une CMC. La figure du haut représente la CMC p2D. La figure du bas donne la CMC équivalente. On remarquera la présence de transitions entre le dernier état de la CMC précédente et le premier état de la CMC suivante (notées p1). Notons également les transitions de rebouclage sur le super-état (notées p2).

5.2. Apprentissage simultané de plusieurs vecteurs d'observations

Précédemment, les données étaient soumises au système sous la forme d'un seul vecteur d'observations rassemblant toutes les données de l'image. Maintenant elles vont être présentées sous forme de plusieurs vecteurs d'observations construits selon la procédure suivante : on déplace une fenêtre d'échantillonnage sur l'image. A chaque position de la fenêtre on va constituer un vecteur. Ce vecteur est simplement réalisé en juxtaposant les valeurs des niveaux de gris de la zone contenue dans la fenêtre, ligne par ligne. Ainsi, une fenêtre de 5×3 pixels engendrera à chaque position sur l'image un vecteur de $5 \times 3 = 15$ valeurs, composé de la juxtaposition de 5 lignes de 3 pixels. On peut également permettre un recouvrement partiel des fenêtres. Cette technique permet de réduire l'effet du découpage arbitraire imposé par le déplacement de la fenêtre. D'autre part elle augmente artificiellement le nombre de données à soumettre à la CMC.

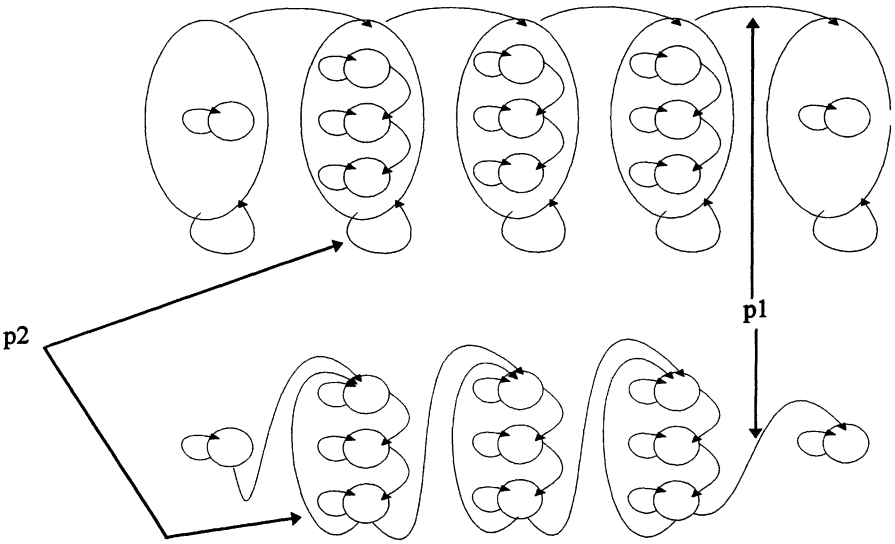


FIGURE 3
Transformation d'une CMC pseudo-2D en une CMC

La phase d'apprentissage consiste à soumettre l'ensemble de ces vecteurs au modèle. Un algorithme de Baum-Welch adapté à cette situation [9] va optimiser les paramètres du modèle (voir ci-dessous). Remarquons qu'il ne s'agit pas ici de faire apprendre à la CMC une sorte de vecteur prototype. Il ne faut pas oublier que l'architecture de la CMC est de type gauche droite, mais que des retours en arrière sont autorisés sur certains groupes d'états. Cela revient à dire qu'un groupe d'états (un super état) va apprendre un certain groupe de vecteurs. Ainsi, un super état se trouve associé à une zone de l'image. C'est l'algorithme d'apprentissage qui détermine la frontière entre les groupes de vecteurs.

5.3. Notations

Une observation o est constituée de K vecteurs o^i, chacun de longueur T^i.

$$o = \{o_1, o_2, \ldots o^{K-1}, o^K\} \text{ avec } o^i = \{o_1^i, o_2^i, \ldots o_{T^i-1}^i, o_{T^i}^i\}$$

On applique le calcul des $\alpha_t(i)$ et des $\beta_{t+1}(j)$ à chaque vecteur d'observations o^k. Ce qui conduit donc au calcul de $\alpha_t^k(i)$, $\beta_{t+1}^k(j)$, et de $P(o^k|\lambda)$.

5.4. Réestimation de la matrice II

On considère maintenant la probabilité de commencer dans l'état s_i sachant les K vecteurs d'observations. On propose ici d'estimer cette probabilité par la moyenne. Il serait, bien sûr, possible d'adopter d'autres stratégies telles que, par exemple, la probabilité maximum, ou la probabilité médiane, ou la probabilité modale...

Soit

$$\gamma_1^k(i) = \frac{\alpha_1^k(i)\beta_1^k(i)}{P(o^k|\lambda)} \quad \forall i \in 1 \ldots N \tag{28}$$

Il vient alors :

$$\overline{\pi}_i = \frac{1}{K}\sum_{k=1}^{K} \frac{\alpha_1^k(i) \cdot \beta_1^k(i)}{P(o^k|\lambda)} \quad \forall i \in 1 \ldots N \tag{29}$$

On peut vérifier que la matrice obtenue est stochastique. Il suffit de calculer la somme des éléments de la matrice réestimée en tenant compte du fait que :

$$\sum_{i=1}^{N} \alpha_1^k(i)\beta_1^k(i) = P(o^k|\lambda) \tag{30}$$

5.5. Réestimation de la matrice A

On considère les variables $\xi_t(i,j)$ et $\gamma_t(i)$ définies dans [18] :

$$\xi_t(i,j) = \frac{\alpha_t(i) \cdot a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j)}{P(o|\lambda)} \quad \begin{matrix} \forall i \in 1 \ldots N \\ \forall j \in 1 \ldots N \end{matrix} \tag{31}$$

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \quad \begin{matrix} \forall i \in 1 \ldots N \\ \forall t \in 1 \ldots T-1 \end{matrix} \tag{32}$$

$$\gamma_t(i) = \frac{\alpha_t(i)}{P(o|\lambda)} \quad \begin{matrix} \forall i \in 1 \ldots N \\ t = T \end{matrix} \tag{33}$$

On étend alors ces expressions en introduisant les K vecteurs :

$$\xi_t^k = (i,j) = \frac{\alpha_t^k(i) \cdot a_{ij} \cdot b_j(o_{t+1}^k) \cdot \beta_{t+1}^k(j)}{P(o^k|\lambda)} \quad \begin{matrix} \forall i \in 1 \ldots N \\ \forall j \in 1 \ldots N \\ \forall k \in 1 \ldots K \end{matrix} \tag{34}$$

$$\gamma_t^k(i) = \sum_{j=1}^{N} \xi_t^k(i,j) \quad \begin{matrix} \forall i \in 1 \ldots N \\ \forall t \in 1 \ldots T^k - 1 \\ \forall k \in 1 \ldots K \end{matrix} \tag{35}$$

$$\gamma_t^k(i) = \frac{\alpha_t^k(i)}{P(o^k|\lambda)} \quad \begin{matrix} \forall i \in 1 \ldots N \\ t = T^k \\ \forall k \in 1 \ldots K \end{matrix} \tag{36}$$

La réestimation est obtenue en sommant sur les K vecteurs de l'observation. Les nouveaux termes sont donc :

$$\overline{a}_{ij} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T^k-1} \xi_t^k(i,j)}{\sum_{k=1}^{K} \sum_{t=1}^{T^k-1} \gamma_t^k(i)} \qquad \begin{matrix} \forall i \in 1 \ldots N \\ \forall j \in 1 \ldots N \end{matrix} \tag{37}$$

On développe le numérateur et le dénominateur :

$$\overline{a}_{ij} = \frac{\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)} \sum_{t=1}^{T^k-1} \alpha_t^k(i) \cdot a_{ij} \cdot b_j(o_{t+1}^k) \cdot \beta_{t+1}^k(j)}{\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)} \sum_{t=1}^{T^k-1} \sum_{h=1}^{N} \alpha_t^k(i) \cdot a_{ih} \cdot b_h(o_{t+1}^k) \cdot \beta_{t+1}^k(h)} \qquad \begin{matrix} \forall i \in 1 \ldots N \\ \forall j \in 1 \ldots N \end{matrix} \tag{38}$$

La matrice A obtenue est stochastique.

5.6. Réestimation de la matrice B

En tenant compte des mêmes remarques qu'au paragraphe 5.5 le terme générique est donné par la relation :

$$\overline{b}_j(s) = \frac{\sum_{k=1}^{K} \left[\sum_{t=1 \cap o_t^k = v_s}^{T^k} \gamma_t^k(j) \right]}{\sum_{k=1}^{K} \sum_{t=1}^{T^k} \gamma_t^k(j)} \qquad \begin{matrix} \forall j \in 1 \ldots N \\ \forall s \in 1 \ldots M \end{matrix} \tag{39}$$

On développe le numérateur et le dénominateur avec :

$$\gamma_t^k(j) = \frac{\alpha_t^k(j) \cdot \beta_t^k(j)}{P(o^k|\lambda)} \qquad \begin{matrix} \forall j \in 1 \ldots N \\ \forall t \in 1 \ldots T^k \\ \forall k \in 1 \ldots K \end{matrix} \tag{40}$$

ce qui donne :

$$\overline{b}_j(s) = \frac{\sum_{k=1}^{K} \left[\sum_{t=1 \cap o_t^k = v_s}^{T^k} \frac{\alpha_t^k(i) \cdot \beta_t^k(i)}{P(o^k|\lambda)} \right]}{\sum_{k=1}^{K} \left[\sum_{t=1}^{T^k} \frac{\alpha_t^k(i) \cdot \beta_t^k(i)}{P(o^k|\lambda)} \right]}$$

$$= \frac{\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)} \left[\sum_{t=1 \cap o_t^k = v_s}^{T^k} \alpha_t^k(i) \cdot \beta_t^k(i) \right]}{\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)} \left[\sum_{t=1}^{T^k} \alpha_t^k(i) \cdot \beta_t^k(i) \right]} \qquad \begin{matrix} \forall j \in 1 \ldots N \\ \forall s \in 1 \ldots M \end{matrix} \tag{41}$$

La matrice B obtenue est stochastique.

5.7. Algorithme de Baum-Welch avec vecteurs d'observations

Cet algorithme implémente les formules de réestimation d'une CMC présentées en (29), (38) et (41). Il s'appuie sur le calcul des variables Forward (12) et Backward (23).

Notations employées dans l'algorithme :

→ K : nombre de vecteurs dans l'observation o

→ NombreIteration : nombre d'itérations à effectuer

→ α^k, β^k et $P(o^k|\lambda)$: matrices α, β et probabilité $P(o|\lambda)$ associées à un vecteur d'observations o^k.

→ calcul_forward_backward() : fonction qui calcule les K matrices α^k (renvoyées dans un tableau de matrices noté α), les K matrices β^k (renvoyées dans un tableau de matrices noté β), et les K valeurs $P(o^k|\lambda)$ (renvoyées dans un tableau de valeurs noté P) d'après une CMC λ et une observation o. Elle utilise la relation vue en 5.4 pour calculer et renvoyer la valeur de $P(o|\lambda)$.

Algorithme : Baum-Welch avec vecteurs d'observations

Début
compteur ← 1;
$P(o|\lambda) \leftarrow 0$;
Tant que ((compteur ⩽ NombreIteration) *et* ($P(o|\lambda) < 1$)) *faire*
/* calculer les α^k, β^k et $P(o^k|\lambda)$ */
$P(o|\lambda)$ ← calcul_forward_backward ($\lambda, o, \alpha, \beta, P$)

/* réestimer les paramètres du modèle */

$$\pi_i^* = \frac{1}{K} \sum_{k=1}^{K} \left[\frac{\alpha_1^k(i) \cdot \beta_1^k(i)}{P(o^k|\lambda)} \right] \qquad \forall i \in 1 \ldots N$$

$$a_{ij}^* = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T^k-1} \alpha_t^k(i) \cdot a_{ij} \cdot b_j(o_{t+1}^k) \cdot \beta_{t+1}^k(j)}{\sum_{k=1}^{K} \sum_{t=1}^{T^k-1} \sum_{r=1}^{N} \alpha_t^k(i) \cdot a_{ir} \cdot b_r(o_{t+1}^k) \cdot \beta_{t+1}^k(r)} \qquad \begin{array}{l} \forall i \in 1 \ldots N \\ \forall j \in 1 \ldots N \end{array}$$

$$b_j^*(s) = \frac{\sum_{k=1}^{K} \left[\sum_{t=1 \wedge o_t^k = v_s}^{T^k} \alpha_t^k(j) \cdot \beta_t^k(j) \right]}{\sum_{k=1}^{K} \sum_{t=1}^{T^k} \alpha_t^k(j) \cdot \beta_t^k(j)} \qquad \begin{array}{l} \forall j \in 1 \ldots N \\ \forall s \in 1 \ldots M \end{array}$$

/* recopier la CMC réestimée λ^* dans la CMC courante λ */

$\lambda \leftarrow \lambda^*$

/* incrémenter le nombre d'itérations */

compteur $\leftarrow$ compteur $+1$;

Fin tant que

Renvoyer (λ^*);

Fin

6. Rescaling appliqué aux formules de réestimation d'une CMC avec vecteurs d'observations

6.1. Notations

- les valeurs sans rescaling sont notées $\alpha_t^k(i), \beta_t^k(i), a_{ij}, b_j(o_t), \pi_i$
- les valeurs avec rescaling sont notées $\tilde{\alpha}_t^k(i), \tilde{\beta}_t^k(i)$
- les coefficients de rescaling sont notés C_t^k, D_t^k
- les réestimations des paramètres sans rescaling de la CMC sont notées $\overline{a}_{ij}, \overline{b}_j(o_t), \overline{\pi}_i$
- les réestimations des paramètres avec rescaling de la CMC sont notées ${}^r\overline{a}_{ij}, {}^r\overline{b}_j(o_t), {}^r\overline{\pi}_i$

6.2. Réestimation de la matrice Π

L'équation (29) devient après introduction des $\tilde{\alpha}_t^k(i), \tilde{\beta}_t^k(i)$:

$$^r\overline{\pi}_i = \frac{1}{K} \sum_{k=1}^{K} \frac{\tilde{\alpha}_1^k(i) \cdot \tilde{\beta}_1^k(i)}{P(o^k|\lambda)} \qquad \forall i \in 1 \ldots N \tag{42}$$

On a les relations suivantes :

$$\tilde{\alpha}_t^k(i) = C_t^k \cdot \alpha_t^k(i) \qquad \forall i \in 1 \ldots N \tag{43}$$

$$\tilde{\beta}_t^k(i) = D_t^k \cdot \beta_t^k(i) \qquad \forall i \in 1 \ldots N \tag{44}$$

Compte tenu des relations (43) et (44), l'équation (42) devient :

$$^r\overline{\pi}_i = \frac{1}{K} \sum_{k=1}^{K} \frac{C_1^k \cdot \alpha_1^k(i) \cdot D_1^k \cdot \beta_1^k(i)}{P(o^k|\lambda)} \quad \forall i \in 1 \ldots N \tag{45}$$

En considérant que :

$$D_1^k = \left(\prod_{s=1}^{T^k} c_s^k \right) = \frac{1}{P(o^k|\lambda)} \tag{46}$$

on se ramène à :

$$^r\overline{\pi}_i = \frac{1}{K} \sum_{k=1}^{K} \left[\frac{\alpha_1^k(i) \cdot \beta_1^k(i)}{P(o^k|\lambda)} \cdot \frac{C_1^k}{P(o^k|\lambda)} \right] \quad \forall i \in 1 \ldots N \tag{47}$$

Nous devons introduire un facteur correctif dans (47) afin de retrouver l'équation de réestimation (29). Ici, le facteur dépend de chaque observation. D'où en appliquant cette méthode :

$$\overline{\pi}_i = \frac{1}{K} \sum_{k=1}^{K} \left[\frac{\tilde{\alpha}_1^k(i) \cdot \tilde{\beta}_1^k(i)}{C_1^k} \right] \quad \forall i \in 1 \ldots N \tag{48}$$

La matrice obtenue est stochastique.

6.3. *Réestimation de la matrice* A

L'équation (38) devient après introduction des $\tilde{\alpha}_t^k(i), \tilde{\beta}_t^k(i)$:

$$ {}^r\overline{a}_{ij} = \frac{\displaystyle\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)} \sum_{t=1}^{T^k-1} \tilde{\alpha}_t^k(i) \cdot a_{ij} \cdot b_j(o_{t+1}^k) \cdot \tilde{\beta}_{t+1}^k(j)}{\displaystyle\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)} \sum_{t=1}^{T^k-1} \sum_{l=1}^{N} \tilde{\alpha}_t^k(i) \cdot a_{il} \cdot b_l(o_{t+1}^k) \cdot \tilde{\beta}_{t+1}^k(l)} \quad \begin{matrix} \forall i \in 1 \ldots N \\ \forall j \in 1 \ldots N \end{matrix} \tag{49} $$

Considérant les équations (43) et (44), il vient :

$$ {}^r\overline{a}_{ij} = \frac{\displaystyle\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)} \sum_{t=1}^{T^k-1} C_t^k \cdot \alpha_t^k(i) \cdot a_{ij} \cdot b_j(o_{t+1}^k) \cdot D_{t+1}^k \cdot \beta_{t+1}^k(j)}{\displaystyle\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)} \sum_{t=1}^{T^k-1} \sum_{l=1}^{N} C_t^k \cdot \alpha_t^k(i) \cdot a_{il} \cdot b_l(o_{t+1}^k) \cdot D_{t+1}^k \cdot \beta_{t+1}^k(l)} \quad \begin{matrix} \forall i \in 1 \ldots N \\ \forall j \in 1 \ldots N \end{matrix} \tag{50} $$

On simplifie cette équation en considérant que :

$$ C_t^k \cdot D_{t+1}^k = \left(\prod_{s=1}^{t} c_s^k \right) \cdot \left(\prod_{s=t+1}^{T^k} c_s^k \right) = \left(\prod_{s=1}^{T^k} c_s^k \right) = C_{T^k}^k = \frac{1}{P(o^k|\lambda)} \quad \forall k \in 1 \ldots K \tag{51} $$

Ce qui donne :

$$ {}^r\overline{a}_{ij} = \frac{\displaystyle\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)^2} \sum_{t=1}^{T^k-1} \alpha_t^k(i).a_{ij} \cdot b_j(o_{t+1}^k) \cdot \beta_{t+1}^k(j)}{\displaystyle\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)^2} \sum_{t=1}^{T^k-1} \sum_{l=1}^{N} \alpha_t^k(i) \cdot a_{il} \cdot b_l(o_{t+1}^k) \cdot \beta_{t+1}^k(l)} \quad \begin{matrix} \forall i \in 1 \ldots N \\ \forall j \in 1 \ldots N \end{matrix} \tag{52} $$

Il suffit de corriger l'expression en supprimant le facteur $P(o^k|\lambda)^{-1}$ de l'équation (49) car ce facteur apparaît naturellement à cause du rescaling. Ce qui

donne :

$$\overline{a}_{ij} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T^k-1} \tilde{\alpha}_t^k(i) \cdot a_{ij} \cdot b_j(o_{t+1}^k) \cdot \tilde{\beta}_{t+1}^k(j)}{\sum_{k=1}^{K} \sum_{t=1}^{T^k-1} \sum_{l=1}^{N} \tilde{\alpha}_t^k(i) \cdot a_{il} \cdot b_l(o_{t+1}^k) \cdot \tilde{\beta}_{t+1}^k(l)} \qquad \begin{matrix} \forall i \in 1 \ldots N \\ \forall j \in 1 \ldots N \end{matrix} \tag{53}$$

La matrice obtenue est stochastique.

6.4. Réestimation de la matrice B

L'équation (41) devient après introduction des $\tilde{\alpha}_t^k(i), \tilde{\beta}_t^k(i)$:

$${}^r\overline{b}_j(l) = \frac{\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)} \sum_{t=1 \cap o_t^k = v_l}^{T^k} \tilde{\alpha}_t^k(i) \cdot \tilde{\beta}_t^k(i)}{\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)} \sum_{t=1}^{T^k} \tilde{\alpha}_t^k(i) \cdot \tilde{\beta}_t^k(i)} \qquad \begin{matrix} \forall j \in 1 \ldots N \\ \forall l \in 1 \ldots M \end{matrix} \tag{54}$$

Considérant les équations (43) et (44), il vient :

$${}^r\overline{b}_j(l) = \frac{\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)} \sum_{t=1 \cap o_t^k = v_l}^{T^k} C_t^k \cdot \alpha_t^k(j) \cdot D_t^k \cdot \beta_t^k(j)}{\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)} \sum_{t=1}^{T^k} C_t^k \cdot \alpha_t^k(j) \cdot D_t^k \cdot \beta_t^k(j)} \qquad \begin{matrix} \forall j \in 1 \ldots N \\ \forall l \in 1 \ldots M \end{matrix} \tag{55}$$

On simplifie cette équation en considérant que :

$$C_t^k \cdot D_t^k = \left(\prod_{s=1}^{t} c_s^k \right) \cdot \left(\prod_{s=t}^{T^k} c_s^k \right) = c_t^k \cdot C_{T^k}^k = \frac{c_t^k}{P(o^k|\lambda)} \qquad \forall j \in 1 \ldots N \tag{56}$$

Ce qui donne :

$${}^r\overline{b}_j(l) = \frac{\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)^2} \sum_{t=1 \cap o_t^k = v_l}^{T^k} c_t^k \cdot \alpha_t^k(j) \cdot \beta_t^k(j)}{\sum_{k=1}^{K} \frac{1}{P(o^k|\lambda)^2} \sum_{t=1}^{T^k} c_t^k \cdot \alpha_t^k(j) \cdot \beta_t^k(j)} \qquad \begin{matrix} \forall j \in 1 \ldots N \\ \forall l \in 1 \ldots M \end{matrix} \tag{57}$$

Il suffit de corriger l'expression en supprimant le facteur $P(o^k|\lambda)^{-1}$ de l'équation (54) car ce facteur apparaît naturellement à cause du rescaling. Il faut également introduire un facteur pour corriger l'effet du rescaling sur les produits $\tilde{\alpha}_t^k(j) \cdot \tilde{\beta}_t^k(j)$. Ce qui donne :

$$\bar{b}_j(l) = \frac{\sum_{k=1}^{K} \left[\sum_{t=1 \cap o_t^k = v_l}^{T^k} \frac{\tilde{\alpha}_t^k(j) \cdot \tilde{\beta}_t^k(j)}{c_t^k} \right]}{\sum_{k=1}^{K} \sum_{t=1}^{T^k} \frac{\tilde{\alpha}_t^k(j) \cdot \tilde{\beta}_t^k(j)}{c_t^k}} \qquad \begin{array}{l} \forall j \in 1 \ldots N \\ \forall l \in 1 \ldots M \end{array} \tag{58}$$

La matrice B obtenue est stochastique.

7. Application à l'apprentissage d'images

Nous présentons ici une illustration des méthodes présentées précédemment à l'apprentissage d'images. Nous avons réalisé ces tests sur des images (100 × 100 pixels, 256 niveaux de gris). Chaque image est transformée en observation de deux manières. La première consiste à former un seul vecteur avec tous les points de l'image. La seconde consiste à échantillonner l'image par une fenêtre mobile, générant ainsi un vecteur à chaque nouvelle position de la fenêtre. Dans le premier cas, on obtient un seul vecteur de 10000 valeurs. Dans le second cas nous avons obtenu 400 vecteurs de 25 valeurs chacun (les fenêtres avaient une taille de 5x5, et sur chaque axe on peut former 100/5 = 20 fenêtres). Puis nous avons traité ces observations par chacun des algorithmes d'optimisation. La première observation (l'unique vecteur de 10000 valeurs) a été fournie à deux algorithmes : Baum-Welch sans rescaling (BWSR) et Baum-Welch avec rescaling (BWAR). Le second ensemble d'observations (les 400 vecteurs de 25 valeurs) a été fourni aux algorithmes de Baum-Welch adaptés aux vecteurs d'observations sans rescaling ($BW_{VO}SR$) et avec rescaling ($BW_{VO}AR$). On a préalablement déterminé une CMC pseudo-2D d'après l'image, puis converti celle-ci en CMC. Cette CMC sert de point de départ pour les quatre algorithmes. Nous avons paramétré l'algorithme de construction de la CMC p2D pour avoir peu d'états cachés (La CMC obtenue comporte neuf états cachés).

A l'issue de ces tests nous avons compté dans la matrice α le nombre de valeurs supérieures à un certain seuil (respectivement 10^{-2} et 10^{-4}) car la probabilité $P(O|\lambda)$ est calculée à partir des coefficients de cette matrice. Les valeurs inférieures à ce seuil sont considérées comme nulles. On obtient ainsi deux classes. La première est celle des valeurs de probabilités nulles (inférieures au seuil de 10^{-2} ou 10^{-4}) et la seconde est celle des valeurs non-nulles. Les tableaux et les graphiques suivants présentent les résultats obtenus pour 6 images en appliquant les quatre algorithmes.

Le tableau 1 donne pour chacune des six images testées le pourcentage de valeurs non-nulles (supérieures au seuil de 10^{-2}) contenues dans la matrice α à l'issue de l'optimisation avec chacun des quatre algorithmes. Le tableau 2 donne les mêmes renseignements au seuil de 10^{-4}. Dans ce dernier, on constate, qu'en moyenne l'emploi de l'algorithme de Baum-Welch sans rescaling avec un seul vecteur

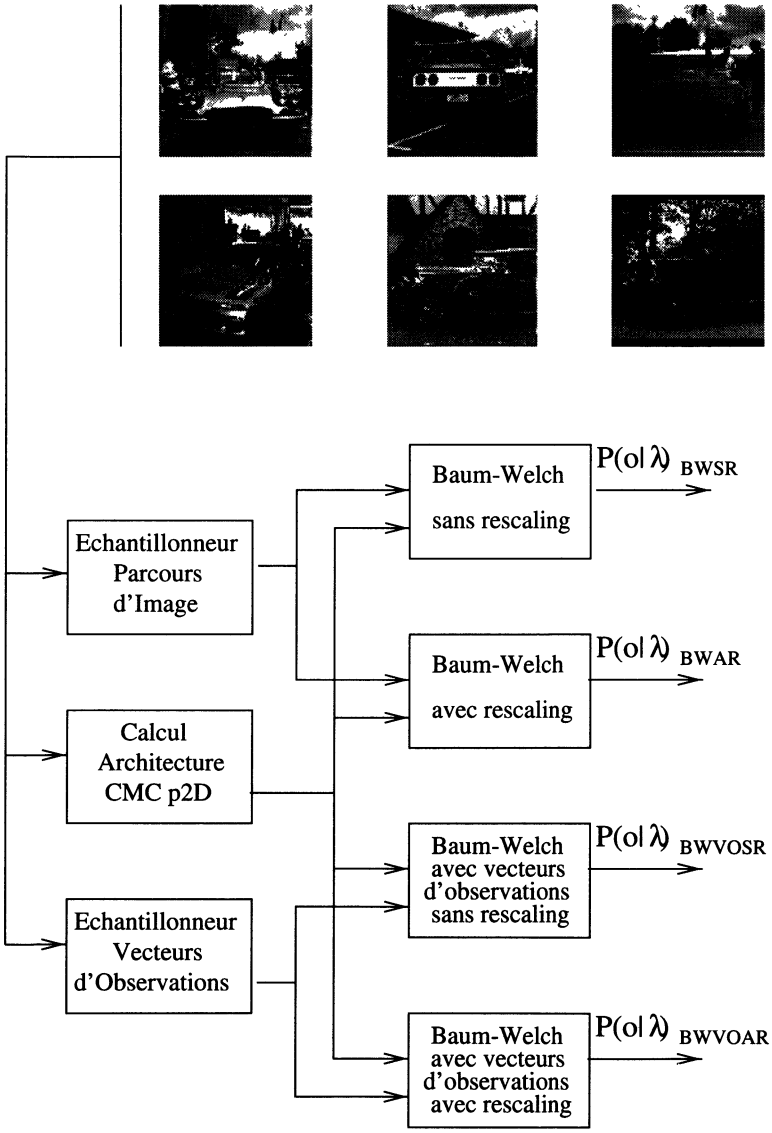


FIGURE 4
Schématisation du test de comparaison réalisé

TABLEAU 1
Pourcentage de valeurs non-nulles dans la matrice α pour chaque algorithme (au seuil de 10^{-2}).

images	BWSR	BWAR	BW$_{VO}$SR	BW$_{VO}$AR
image 1	0	70.62	2.15	70.44
image 2	0	66.45	1.41	67.71
image 3	0	63.20	2.92	62.20
image 4	0	60.41	1.47	59.70
image 5	0	64.85	1.48	64.86
image 6	0	59.21	1.39	58.23
moyenne	0	64.12	1.80	63.92

TABLEAU 2
Pourcentage de valeurs non-nulles dans la matrice α pour chaque algorithme (au seuil de 10^{-4})

images	BWSR	BWAR	BW$_{VO}$SR	BW$_{VO}$AR
image 1	0.014	95.98	9.99	95.77
image 2	0.004	96.12	8.90	95.96
image 3	0.012	98.22	20.50	97.86
image 4	0.012	98.31	10.76	98.07
image 5	0	97.85	9.71	97.83
image 6	0.004	96.69	11.92	96.81
moyenne	0.007	97.19	11.96	97.05

d'observations ne fait apparaître dans la matrice α que 0.7% de valeurs non nulles. L'ajout d'une technique de rescaling porte ce résultat à 97.19%. L'emploi de la technique des vecteurs d'observations sans rescaling obtient un résultat (11.96%), bien que très faible, nettement meilleur par rapport à la méthode de base (sans rescaling, ni vecteurs d'observations). L'ajout de la technique de rescaling donne un résultat sensiblement équivalent à la méthode qui n'emploie qu'un seul vecteur d'observations (97.05%). Soulignons ici le fait que les méthodes qui n'emploient qu'un seul vecteur d'observations n'ont pu apprendre la CMC. Les deux algorithmes sont en effet sujet au problème d'overflow évoqué au paragraphe 4.5. Par ailleurs, il est facile de constater sur les figures 5 et 6 que le rescaling produit des résultats plus stables en fonction des images que les méthodes sans rescaling. Notons également qu'entre les seuils de 10^{-2} et de 10^{-4}, le phénomène d'amplification reste encore très marqué.

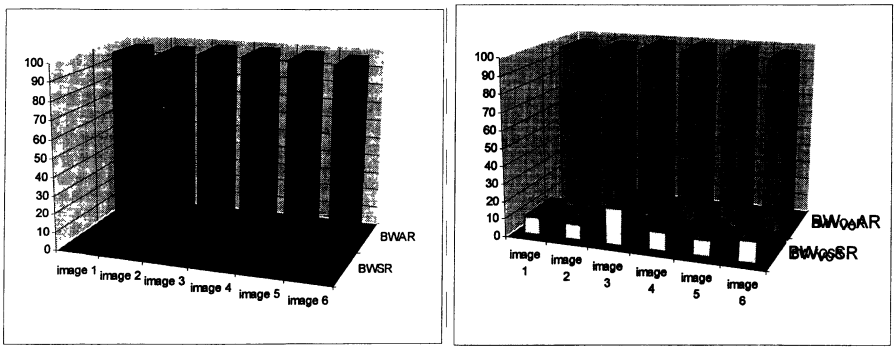


FIGURE 5

Pourcentage de valeurs non-nulles dans la matrice α dans le cas d'une optimisation avec et sans rescaling (cas avec un seul vecteur d'observations à gauche et cas avec plusieurs vecteurs d'observations à droite) au seuil de 10^{-4}.

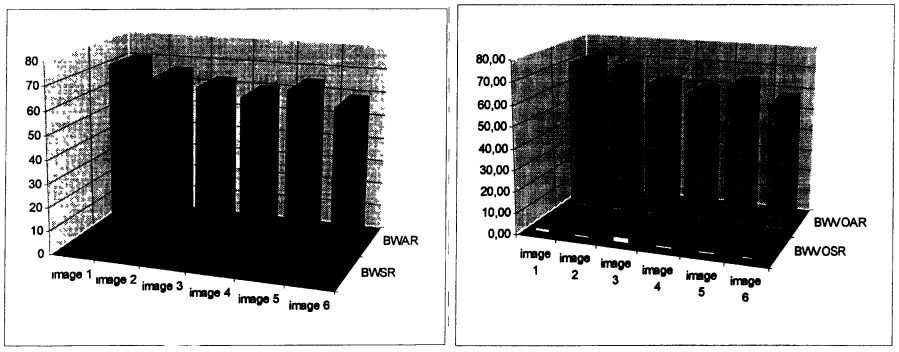


FIGURE 6

Pourcentage de valeurs non-nulles dans la matrice α dans le cas d'une optimisation avec et sans rescaling (cas avec un seul vecteur d'observations à gauche et cas avec plusieurs vecteurs d'observations à droite) au seuil de 10^{-2}.

8. Conclusion

Dans cet article on a présenté les principaux problèmes numériques rencontrés lors de la mise en œuvre des chaînes de Markov cachées. On montre, plus particulièrement, que ces difficultés ont de fortes chances de se produire lorsque l'on désire utiliser cet outil pour la modélisation ou l'apprentissage d'image numériques. Constatant que ces problèmes n'avaient pas reçu, jusqu'à aujourd'hui, de solution satisfaisante nous avons proposé dans ce travail des méthodes et des algorithmes de rescaling des probabilités susceptibles d'être utilisés pour le traitement des images numériques. Les expérimentations réalisées sur six images montrent les améliorations remarquables obtenues par cette approche par rapport à des solutions plus classiques basées soit sur un seul vecteur d'observations soit sur des algorithmes sans rescaling.

9. Bibliographie

[1] AGAZZI O., KUO S.S., (1994), Keyword spotting in poorly printed documents using pseudo-2D HMMs, IEEE Transactions on pattern recognition and machine intelligence, vol. 16(8), pp 842-848.

[2] ALANI T., GUELLIF H., (1994), Modèles de Markov cachés – théorie et techniques de base – Partie 1, Rapport de recherche n°2196, INRIA, 57 p.

[3] ANIGBOGU J.-C., (1992), Reconnaissance de caractères imprimés multifontes à l'aide de modèles stochastiques et métriques, Thèse de doctorat, Université de Nancy I, 156 p.

[4] ASSELIN de BEAUVILLE J.-P., SLIMANE M., VENTURINI G., LAPORTE J.-L., NARBEY M., (1996), Two hybrid gradient and genetic search algorithms for learning hidden Markov models, Proceedings of the International Conference on Machine Learning (ICML'96), Bari (Italy), pp 5-12.

[5] ASKAR M., DERIN H., (1981), A recursive algorithm for the Bayes solution of the smoothing problem, IEEE Transactions on Automatic Control, vol. 26(2), pp 558-561.

[6] BAUM L.E., EAGON J.A., (1967), An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology, Bull. Amer. Soc.73, pp 360-363.

[7] BAUM L.E., (1972), A inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes, Inequalities 3, pp 1-8.

[8] BROUARD T., SLIMANE M., ASSELIN de BEAUVILLE J.-P., (1996), Les modèles de Markov cachés pseudo-2D, Rapport Interne n°176, LI EIII, Tours, 26 p.

[9] BROUARD T., SLIMANE M., ASSELIN de BEAUVILLE J.-P., (1996), Introduction aux modèles de Markov cachés du 1er ordre (2ème Partie), Rapport Interne n°178, LI EIII, Tours, 68 p.

[10] BROUARD T., SLIMANE M., VENTURINI G., ASSELIN de BEAUVILLE J.-P., (1997), Apprentissage du nombre d'états d'une chaîne de Markov cachée pour la reconnaissance d'images, 16ème Colloque GRETSI sur le Traitement du Signal et des Images, Grenoble, pp. 845-848.

[11] CHEN J.-L., KUNDU A., (1994), Rotation and gray scale transform invariant texture identification using wavelet decomposition and hidden Markov model, IEEE Transactions on pattern analysis and machine intelligence, vol. 16(2), pp 208-215.

[12] DEVIJVER P.A., DEKESEL M., (1988), Champs aléatoires de Pickard et modélisation d'images digitales, Traitement du Signal, vol. 5(5), pp 131-150.

[13] DOURS C., (1989), Contribution à l'étude du décodage acoustico-phonétique pour la reconnaissance automatique de la parole, Thèse de Doctorat, Univ. Paul Sabatier, Toulouse.

[14] GAUVAIN J.-L., LAMEL L.F., ADDA G., ADDA-DECKER M., (1994), The LIMSI continuous speech dictation system : Evaluation on the ARPA Wall Street Journal Task, Proceeding of ICCASP'94, pp 1557-1560.

[15] KRIOUILLE A., (1990), La reconnaissance automatique de la parole et les modèles markoviens cachés, Thèse de Doctorat, Université de Nancy I, 149 p.

[16] LEVINSON S.E., RABINER L.R., SONDHI M.M., (1983), An introduction to the application of the theory of probabilistic functions of Markov process to automatic speech recognition, The Bell System Technical Journal, vol. 62(4).

[17] MAO W.D., KUNG S.Y., (1990), An object recognition system using stochastic knowledge source and VLSI architecture, Proceedings of the International Conference on Pattern Recognition, pp 832-836.

[18] RABINER L.R., (1989), A tutorial on hidden Markov models and selected application in speech recognition, Proceedings of IEEE, vol. 77, pp 257-286.

[19] RABINER L.R., LEVINSON S.E., SONDHI M.M., (1983), On the application of vector quantization and hidden Markov models to speaker-independant isolated word recognition, The Bell System Technical Journal, vol. 62, pp 1075-1105.

[20] SAERENS M., (1993), Hidden Markov models assuming a continuous time dynamic emission of acoustic vectors, Proceedings of Eurospeech.

[21] SALZENSTEIN F., (1996), Modèles Markoviens flous et segmentation statistique non supervisée d'image, Thèse de doctorat, Université de Rennes I, 142 p.

[22] SAMARIA F.S., (1994), Face recognition using HMMs, Dissertation for the degree of Doctor of Philosophy, Trinity College, University of Cambridge, 101 p.

[23] SAMARIA F.S., FALLSIDE F., (1993), Face identification and features extract using HMMs, Image processing : theory and applications, Elsevier Science Publisher B.V, pp 295-298.

[24] SLIMANE M., ASSELIN de BEAUVILLE J.-P., (1994), Introduction aux modèles de Markov cachés du premier ordre (1ère Partie), Rapport interne n°171, LI EIII, Tours, 36 p.

[25] SLIMANE M., VENTURINI G., ASSELIN de BEAUVILLE J.-P., BROUARD T., BRANDEAU A., (1996), Optimizing HMM with a genetic algorithm, Artificial Evolution, Lecture Notes in Computer Science, vol. 1063, Springer Verlag, pp 384-396.

[26] SLIMANE M., ASSELIN de BEAUVILLE J.-P., BROUARD T., VENTURINI G., SEALELLI J.-M., (1996), Reconnaissance d'image par chaîne de Markov cachée optimisée génétiquement, Actes des XXVIIIèmes Journées de Statistique, Québec, pp 683-687.