

Apprentissage du nombre d'états d'une chaîne de Markov cachée pour la reconnaissance d'images

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

Laboratoire d'Informatique
Ecole d'Ingénieurs en Informatique pour l'Industrie
Université de Tours
64, Avenue Jean Portalis, 37200 Tours (FRANCE)
Tel: (+33)-02-47-36-14-14, Fax: (+33)-02-47-36-14-22
Email : brouard, slimane, venturini, asselin@univ-tours.fr

RÉSUMÉ

ABSTRACT

L'apprentissage d'images par Chaînes de Markov cachée (C.M.C.) est un problème difficile, principalement à cause de la quantité de données à faire apprendre à la CMC. De nombreux algorithmes d'apprentissage existent dans ce domaine. Cet article présente un algorithme hybride d'apprentissage de CMC. Cet algorithme optimise à la fois le nombre d'états et les paramètres (probabilités) de la CMC. Il repose sur une recherche génétique d'un bon modèle parmi une population hétérogène de CMC (contenant différentes architectures) et une optimisation par un algorithme de gradient (Baum-Welch). Ce nouvel algorithme converge rapidement et permet une amélioration significative de l'apprentissage d'images et du taux de reconnaissance par rapport aux algorithmes traditionnels.

Learning images with hidden Markov models (HMM) is a difficult problem, especially because of the important volume of data to be learned by the HMM. Several algorithms exist for this purpose. This paper presents a hybrid algorithm for learning HMM. This algorithm optimizes both at the same time the number of states and the parameters (probabilities) of HMMs. It relies on a genetic search of a good model among an heterogeneous population of HMMs (with different architectures) and on a local optimization with a gradient algorithm (Baum-Welch). This new algorithm converges rapidly and increases significantly the learning of images and the accuracy of recognition compared to standard algorithms.

1 Introduction

Les chaînes de Markov cachées (CMC) [8][12] sont des outils très utilisés dans de nombreux domaines, tels que la reconnaissance de la parole [6], la reconnaissance de l'écriture [2], le traitement du signal [9] et l'analyse d'images [7][10]. De nombreux algorithmes d'apprentissage d'une CMC existent. Ils sont basés, généralement, sur des méthodes de gradient tel que l'algorithme de Baum-Welch (BW) [4]. Deux problèmes majeurs se posent lors de leur utilisation. Il s'agit d'une part d'éviter les maxima locaux dans la fonction d'évaluation des CMC et d'autre part de déterminer le bon nombre d'états cachés des chaînes apprises. Dans nos travaux précédents, nous avons résolu efficacement le premier problème grâce à une exploration parallèle de l'espace des paramètres utilisant BW comme opérateur local de gradient dans un algorithme génétique (AG) [3][13]. Le second problème correspond à un thème générique en reconnaissance des formes, c'est-à-dire à la détermination d'une bonne architecture, problème que l'on rencontre notamment avec les réseaux de neurones artificiels mais qui est encore peu étudié en particulier dans les applications des CMC en traitement d'images.

(l'algorithme résultant étant noté AG \cup BW). Ces algorithmes effectuent un apprentissage plus poussé, permettant de faire apprendre un ensemble de valeurs (appelé observation) à une CMC. Les résultats obtenus ont montré que l'apprentissage hybride était bien meilleur que l'apprentissage traditionnel. Parmi les nombreux paramètres du problème, il en est un qui est généralement fixé a priori, souvent empiriquement, et qui joue un grand rôle dans l'apprentissage, autant en qualité qu'en temps de calcul. Il s'agit du nombre d'états cachés de la CMC. Lors de l'apprentissage d'images, l'architecture de la CMC était déterminée empiriquement [11] par certaines règles appliquées par l'utilisateur. Mais des tests effectués dans ce même travail ont révélés que parfois ces règles admettaient des exceptions. L'algorithme que nous proposons ici suit une autre approche. Cet algorithme génétique hybride optimise simultanément l'architecture et les paramètres d'une CMC en maintenant dans la même population des CMC ayant des nombres d'états cachés différents. Auparavant, l'optimisation ne portait que sur les paramètres de la CMC et le nombre d'états devait être fixé a priori. Ce nouvel algorithme apporte donc une solution au problème de l'apprentissage non supervisé (nombre d'états inconnu) d'une CMC.

Les nouveaux algorithmes introduits dans [3][13] sont nés de l'hybridation d'un AG et d'un algorithme de type gradient

2 Principe de l'algorithme

L'algorithme GHOSP (Genetic Hybrid Optimization & Search of Parameters), dérive de l'algorithme AGUBW [3] adapté à l'image, et emploie les CMC pseudo-2D [1]. En effet, il a été montré que ces CMC sont plus adaptées à la modélisation d'images que les CMC traditionnelles [11]. Le principe de l'algorithme est le suivant : un AG manipule une population de CMC constituée d'individus dont l'architecture n'est pas unique. Par exemple, des CMC ayant de 4 à 12 états. L'algorithme va chercher à obtenir des CMC optimales, c'est à dire des CMC ayant la plus forte probabilité d'engendrer une observation donnée O. Cette probabilité peut être calculée rapidement par l'algorithme Forward [4]. L'optimisation locale est réalisée par l'algorithme de BW conjointement à des opérateurs génétiques spécialisés pour les CMC [3] [13]. Parmi ces opérateurs, par exemple, on trouve un opérateur de croisement, qui intègre les contraintes de stochasticité s'appliquant sur les CMC. Il existe également un opérateur de normalisation, afin de rétablir cette même contrainte après la phase de mutation. Pour gérer les différentes architectures de CMC dans la même population, on applique les opérateurs génétiques sur des individus ayant la même architecture. Ainsi, au cours des itérations de l'AG, les CMC les plus performantes deviendront majoritaires et imposeront le meilleur nombre d'états cachés dans toute la population. Nous donnons ci-dessous la trame de l'algorithme GHOSP. On utilise un marquage, nommé « parent » simplement afin de ne traiter que les individus nécessaires durant l'optimisation et l'évaluation.

1 - initialisation

créer une population de taille S, aléatoirement, contenant des architectures de CMC diverses (en nombre d'états). Aucun individu n'est marqué « parent ».

Lire une observation O

2 - optimisation

appliquer sur chaque CMC de la population non marqué « parent » l'algorithme de Baum-Welch à partir de l'observation O

3 - évaluation

appliquer sur chaque individu de la population qui ne porte pas la marque « parent » l'algorithme de Forward, et noter la valeur de probabilité de l'observation. Pour tous ceux qui portaient cette marque, l'enlever.

3 - sélection

parmi tous les individus de la population, en sélectionner un certain nombre $S' < S$, qui seront utilisés comme parents pour régénérer les S-S' autres individus non retenus. La sélection se réalise suivant les meilleurs scores calculés à la phase 3. Chaque individu sélectionné est marqué « parent ».

4 - croisement

pour chaque individu non marqué « parent », sélectionner au hasard deux individus parmi la population de ceux marqués « parent » et les croiser. Le croisement est à un point de coupure, et se réalise entre deux lignes des matrices des CMC. Ceci permet

d'obtenir en retour deux fils, CMC également. On ne conserve qu'un des deux fils, au hasard.

5 - mutation / normalisation

sur chaque individu non marqué « parent » on applique l'opérateur de mutation. Celui-ci consiste à modifier d'une faible quantité aléatoire chaque coefficient des matrices de la CMC. Chaque coefficient est modifié, en fonction de la valeur de la probabilité de mutation. Après avoir traité un individu, on lui applique un opérateur de normalisation, afin que cet individu soit une CMC (matrices stochastiques).

6 - évaluation de la condition d'arrêt

si le nombre d'itérations maximum n'est pas atteint, alors retourner à l'étape 2, sinon aller à l'étape 7.

7 - renvoyer la meilleure CMC parmi la population en cours

3 Le système d'apprentissage

On peut le décomposer en trois phases : codage, apprentissage, enregistrement.

Le rôle de la première phase est d'échantillonner les données d'entrée. On va transformer les données brutes (niveaux de gris des pixels dans notre cas) en observations, qui seront apprises par la CMC. Pour cela on associe à chaque niveau de gris un code. On peut avoir une correspondance directe, c'est à dire qu'à chaque niveau de gris de l'image correspond un code différent. Mais on peut aussi avoir une transformation (linéaire dans notre cas) qui associe un code à une plage de niveaux de gris. Cela revient en fait à diminuer le nombre de niveaux de gris de l'image avant de coder. Ce codage achevé, on distribue les données codées suivant des vecteurs. Pour cela, on définit une fenêtre de taille fixe, qui va glisser sur l'image. A chaque position de cette fenêtre, on crée un vecteur en réorganisant les données contenues dans la fenêtre. Pour cette réorganisation, nous avons simplement considéré les données lignes par lignes. Moyennant des contraintes sur la fenêtre glissante, on pourrait utiliser d'autres parcours d'images tel que celui de Peano par exemple. Le glissement de la fenêtre est défini par un pas sur chaque axe. Le pas peut être inférieur aux dimensions de la fenêtre, ce qui autorise un recouvrement et augmente artificiellement la quantité de données apprises.

La seconde phase est constituée par la mise en œuvre de l'algorithme GHOSP. Celui-ci étant paramétré (tableau 1), il va rechercher dans tout l'espace des CMC qui lui est accessible celle qui a la probabilité maximum de générer l'ensemble des vecteurs d'observations constitué à la phase 1. Notons que cet algorithme a été adapté pour l'optimisation de vecteurs d'observations. La réestimation se fait de telle sorte que la CMC ait une probabilité maximale de générer l'ensemble des vecteurs. Afin de s'affranchir de certains problèmes de calcul (underflow) nous avons utilisé des techniques de rescaling [7] (re-échelonnage des valeurs numériques dans le but d'éviter les dépassements de capacités de calcul de la machine). Au terme de l'exécution de GHOSP, la chaîne de Markov cachée, ainsi que divers paramètres tels que les paramètres d'apprentissage par exemple, sont enregistrés dans une base de données. Cette

base servira pour la phase de reconnaissance. L'enregistrement des données constitue la phase 3. Le schéma du système d'apprentissage est représenté figure 1.

Paramètres d'apprentissage	valeur typique
taille de la population de l'AG	50
nombre de parents conservés	20
probabilité de croisement	1
probabilité de mutation	0.01
nombre d'itérations de l'AG	20
nombre d'itérations de BW	5
largeur de la fenêtre glissante	8
hauteur de la fenêtre glissante	8
glissement suivant l'axe X	6
glissement suivant l'axe Y	6

Tableau 1—Paramètres d'apprentissage de GHOSP

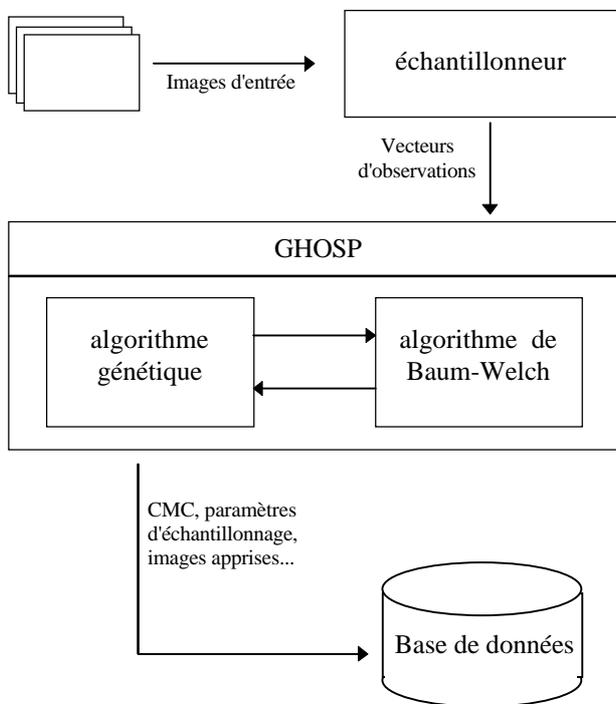


Figure 1—Schéma du système d'apprentissage

4 Le système de reconnaissance

La reconnaissance se fait par modèle discriminant. C'est à dire que l'apprentissage va associer à chaque image apprise une CMC. La reconnaissance se fera en calculant pour chaque CMC connue sa probabilité d'engendrer l'image à reconnaître. On reconnaîtra l'image dont la CMC associée a obtenu un score maximum.

La reconnaissance se déroule en deux temps : évaluation des scores de chaque CMC puis, sélection de la meilleure CMC.

Une phase préliminaire consiste à échantillonner l'image à reconnaître afin de constituer les vecteurs d'observations.

La phase suivante est itérative. Pour chaque CMC de la base de données constituée durant la phase d'apprentissage, on calcule à l'aide de l'algorithmique Forward sa probabilité de

générer l'image présentée en entrée du système de reconnaissance. On obtient pour chaque CMC un ensemble de probabilités. En fait, on en a une pour chaque vecteur d'observations d'une même image, et une, globale, pour l'ensemble des vecteurs de l'image. Cela permet d'utiliser différentes stratégies de décision pour la phase suivante de décision.

La décision choisit la CMC qui a la plus forte probabilité de générer les données d'entrée. On peut utiliser, par exemple, une politique fondée uniquement sur la probabilité de l'ensemble des vecteurs. Une seconde solution serait de mettre en œuvre un algorithme de classification sur les vecteurs de probabilités obtenus à la phase d'évaluation. La CMC désignée comme optimale donne l'image reconnue (qui lui a été associée durant l'apprentissage).

Le schéma du système de reconnaissance est représenté figure 2.

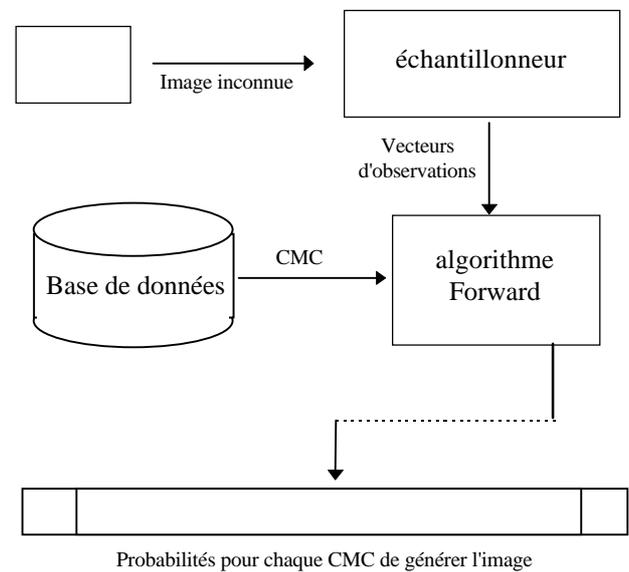


Figure 2—Schéma du système de reconnaissance

5 Résultats

Les résultats suivants (tableaux 2 et 3) montrent les performances moyennes obtenues en phases d'apprentissage et de test de 12 images de 100x100 pixels et comportant 256 niveaux de gris.

Les valeurs indiquées tableau 2 (et figure 4) représentent le $P(O|\lambda)$ moyen (probabilité que la CMC génère l'image) pour l'ensemble des 12 images non bruitées, chaque algorithme ayant été exécuté 10 fois. Remarquons que l'accroissement de $P(O|\lambda)$, avec le nouvel algorithme GHOSP est notable par rapport à l'ancienne version AGUBW. D'autre part, à l'issue de nos tests, le nouvel algorithme trouve toujours une solution meilleure que l'ancien. Le tableau 3 indique les résultats moyens en reconnaissance obtenus à l'issue de la phase de test sur les mêmes images mais bruitées. Les quatre différents bruits testés étant : gaussien, non-corrélé, impulsif et uniforme, ce qui donne $4 \times 12 = 48$ images. Il

apparaît que l'algorithme GHOSP tend à trouver une bonne architecture en peu d'itérations. Une étude est en-cours afin d'isoler le rôle de chaque paramètre de l'apprentissage.

BW	AGuBW	GHOSP
1.61×10^{-5}	7.67×10^{-4}	2.31×10^{-3}

Tableau 2— Résultats moyens des apprentissages.

BW	AGuBW	GHOSP
58.75 %	86.25 %	89.25 %

Tableau 3 — Résultats moyens lors de la reconnaissance des images bruitées.



Figure 3 — Exemples de quelques images d'apprentissage

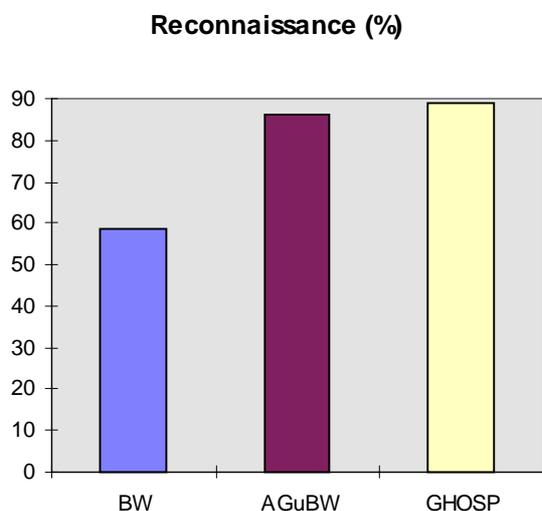


Figure 4 — Résultats moyens lors de la reconnaissance des images bruitées

4 Références

- [1] Agazzi O., "HMM based optical character recognition and the presence of deterministic transformations", Pattern Recognition, 26(12):1813--1826, 1993.
- [2] Anigbogu J.C. "Reconnaissance de textes imprimés multifontes à l'aide de modèles stochastiques et métriques", Thèse de doctorat, Université de Nancy I, 157p, 1992
- [3] Asselin de Beauville J.P., Slimane M., Venturini G., Laporte J.L., Narbey M. "Two hybrid gradient and genetic search algorithms for learning hidden Markov models", ICML'96 Workshop on Evolutionary Computing and Machine Learning, Bari (Italy), pp. 5-12, 1996.
- [4] Baum L.E., Eagon J.A. "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology", Bull American Society, vol 73, pp. 360-363, 1967.
- [5] Brouard T., Slimane M., Asselin de Beauville J.P. "Les modèles de Markov cachés pseudo-2D", Rapport interne 176, Laboratoire d'Informatique, E3i, Tours, 26p., 1996.
- [6] Kriouille A. "La reconnaissance de la parole et les modèles Markoviens cachés", Thèse de doctorat, Université de Nancy I, 149p., 1990.
- [7] Mao W.D., Kung S.Y. "An object recognition system using stochastic knowledge source and VLSI architecture", Proceedings of the International Conference on Pattern Recognition, pp. 832-836, 1990.
- [7] Rabiner L.R. "A tutorial on hidden Markov models and selected applications in speech recognition", Proceedings of IEEE, vol 77, pp. 257-286, 1989.
- [9] Saerens M. "Hidden Markov models assuming a continuous time dynamic emission of acoustic vectors" Proceedings of Eurospeech, sp., 1993.
- [10] Salzenstein F. "Modèles Markoviens flous et segmentation statistique non supervisée d'image", Thèse de doctorat, Université de Rennes I, sp., 1996.
- [11] Samaria F.S., Fallside F. "Face identification and features extract using HMMs", Image processing : theory and applications, Elsevier Science Publisher B.V, pp. 295-298, 1993
- [12] Slimane M., Asselin de Beauville J.P. "Les modèles de Markov cachés du premier ordre (1ère partie)", Rapport interne 171, Laboratoire d'Informatique, E3i, Tours, 36p., 1994.
- [13] Slimane M., Venturini G., Asselin de Beauville J.P., Brouard T., Brandeau A. "Optimizing HMM with a genetic algorithm", Artificial Evolution, Lecture Notes in Computer Science, Springer Verlag, vol 1063, pp.384-396, 1996.