



HAL
open science

Proposition d'un modèle de réseau bayésien dynamique appliqué à la reconnaissance de mots arabes manuscrits

Nabil Ghanmi, Mohamed Ali Mahjoub, Jayech Khelifa, Najoua Essoukri Ben Amara

► To cite this version:

Nabil Ghanmi, Mohamed Ali Mahjoub, Jayech Khelifa, Najoua Essoukri Ben Amara. Proposition d'un modèle de réseau bayésien dynamique appliqué à la reconnaissance de mots arabes manuscrits. JFRB-Journées Francophones sur les Réseaux Bayésiens, May 2012, 2012-05-13, Tunisie. hal-01177143

HAL Id: hal-01177143

<https://hal.science/hal-01177143>

Submitted on 27 Aug 2015

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.

Proposition d'un modèle de réseau bayésien dynamique appliqué à la reconnaissance de mots arabes manuscrits

Nabil Ghanmi, Mohamed Ali Mahjoub, Khlifia Jayech, Najoua Essoukri Ben Amara

Unité de recherche SAGE – Eniso Ecole Nationale d'Ingénieurs de Sousse
Parc Technologique, 4054, Sahloul, Sousse, Tunisie

nabil.ghanmy@gmail.com, medali.mahjoub@ipeim.rnu.tn, jayech_k@yahoo.fr, najoua.benamara@eniso.rnu.tn

Résumé — L'objectif de ce travail est de présenter une approche pour la reconnaissance de l'écriture manuscrite arabe. Notre approche consiste à construire un modèle bidimensionnel en couplant deux chaînes de Markov cachées (MMCs) à l'aide des réseaux bayésiens dynamiques (RBD). Chaque chaîne de Markov étant associée à un flux d'information (horizontal ou vertical) obtenu en balayant l'image du mot manuscrit suivant une direction donnée (horizontale ou verticale). Ainsi, le modèle de couplage proposé permet de prendre en considération conjointement les deux flux d'informations : celui des colonnes de pixels et celui des lignes de pixels.

Mots clés — RBD ; MMC ; Inférence ; Apprentissage ; Algorithme EM ; maximum de vraisemblance ; Algorithme d'Arbre de Jonction

I. INTRODUCTION

Une grande partie des problèmes de reconnaissance de l'écrit est traitée par des méthodes statistiques vu leur capacité de modélisation de la variabilité d'observations, d'intégration de contexte et d'absorption de bruit. Parmi ces approches statistiques, les approches stochastiques et principalement les modèles de Markov cachés (MMC). Les MMCs peuvent être vu comme des automates stochastiques permettant de déterminer la probabilité d'une suite d'observations. Ces modèles ont montré leur adéquation surtout à traiter la parole [1]. Cependant, la parole est un signal mono-dimensionnel alors que les images sont bidimensionnelles. Les recherches sur la reconnaissance de l'écrit se sont ainsi inspirées des travaux de reconnaissance de la parole en adaptant des techniques, telles les MMCs, au problème de reconnaissance de l'écrit. Des adaptations de ces modèles à l'aspect 2D de l'image ont été effectuées avec les modèles pseudo-2D (MMC-Planar) [2], les modèles à champs de Markov et les modèles de fusion de scores ou de données utilisant deux ou plus classificateurs MMCs. Toutefois, les premiers ne sont pas de véritables techniques 2D : l'idée de base des PHMMs est de partager l'image en zones homogènes (selon une direction choisie : horizontale ou verticale) et de

modéliser chaque zone par un MMC-1D, d'où leur qualification pseudo 2D. Les seconds utilisent les régions (ensembles de cliques). Ils semblent donc adaptés à prendre en compte des règles de dépendances locales. Dans [3], une étude comparative entre les modèles de fusion de données et fusion de scores a été effectuée et il a été démontré que les modèles de fusion de données donnent des résultats meilleurs que ceux des modèles de fusion de scores. Dans notre approche nous allons modéliser les dépendances entre les nœuds (variables aléatoires d'états) en créant des interactions entre les deux HMMs. Pour cela nous utilisons le formalisme des Réseaux Bayésiens dynamiques.

Dans ce document, nous allons commencer par présenter les réseaux bayésiens. Ensuite, nous allons présenter le modèle de couplage de deux MMCs (vertical et horizontal) en utilisant les réseaux bayésiens dynamiques, nous développons les algorithmes d'apprentissage et de reconnaissance pour ce modèle, et nous conduisons des expériences pour illustrer le potentiel du modèle proposé.

II. RÉSEAUX BAYESIENS DYNAMIQUES

A. Définition d'un réseau bayésien

Les réseaux bayésiens [4] sont des modèles graphiques probabilistes permettant la représentation des connaissances incertaines et/ou incomplètes sur des phénomènes complexes et le raisonnement à partir de ces informations. Ils sont donc une combinaison entre la théorie des graphes et la théorie des probabilités afin de fournir des outils efficaces pour les calculs de la distribution de probabilités de l'ensemble des variables aléatoires définissant le domaine d'étude. Le principe associé aux réseaux bayésiens est de tenir compte des indépendances conditionnelles pour simplifier la loi jointe découlant du théorème de Bayes généralisé. Un réseau bayésien permet la description qualitative et quantitative des indépendances conditionnelles entre variables.

Description qualitative: graphe acyclique orienté $G = (V, E)$ où V représente l'ensemble des variables aléatoires (V_1, V_2, \dots, V_n) décrivant les événements du domaine, et E est l'ensemble des arcs. Un arc de V_i vers V_j constitue une relation de dépendance entre ces nœuds.

Description quantitative: ensemble des tables de probabilités conditionnelles associées à chaque variable V_i . Ces tables sont fournies soit par un expert du domaine soit par un apprentissage automatique.

D'une façon formelle, un réseau bayésien est défini par :

- un graphe acyclique orienté (DAG) $G = (V, E)$, où V est l'ensemble des nœuds de G , et E l'ensemble des arcs de G .
- un espace probabilisé fini (Ω, Z, p)
- un ensemble de variables aléatoires associées aux nœuds du graphe et définies sur (Ω, Z, p) , tel que :

$$p(V_1, V_2, \dots, V_n) = \prod_{i=1}^n p(V_i / C(V_i))$$

où $C(V_i)$ est l'ensemble des causes (parents) de V_i dans le graphe G .

B. Représentation de réseaux bayésiens dynamiques

Un réseau bayésien dynamique [5] peut être défini comme étant une répétition de réseaux classiques dans lesquels on rajoute un lien causal d'un pas de temps à l'autre. Ils contiennent chacun un certain nombre de variables aléatoires représentant les observations et les états cachés du processus.

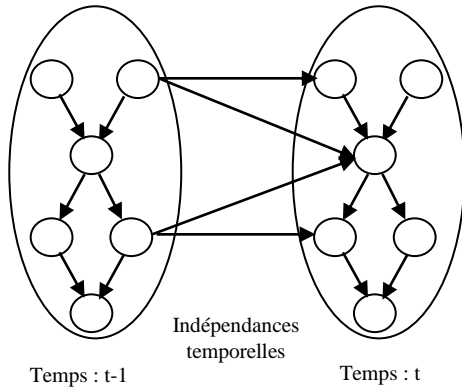


Figure 1. Représentation de deux tranches de temps d'un DBN

On considère un réseau bayésien dynamique composé d'une séquence de T variables d'états cachés (un état caché d'un RBD est représenté par un ensemble de variables d'états cachés) $X = \{x_1, \dots, x_T\}$ et d'une séquence de T variables observables $Y = \{y_1, \dots, y_T\}$ où T est le temps limite du processus étudié.

Pour que la spécification de ce réseau soit complète, nous avons besoin de définir les paramètres suivants :

- La probabilité de transition entre états $P(x_t/x_{t-1})$

- La probabilité conditionnelle d'états cachés sachant une observation $P(y_t/x_t)$
- La probabilité de l'état initial $P(x_1)$

Les deux premiers paramètres doivent être déterminés pour chaque instant $t = 1, \dots, T$. Ces paramètres peuvent être invariant ou non au cours du temps.

Un MMC est un cas particulier de réseau bayésien dynamique [6] :

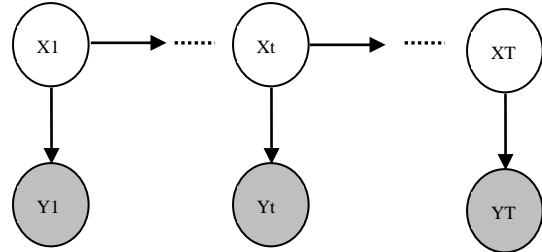


Figure 2. Un MMC représenté comme une instance de RBD déroulé sur 3 pas de temps

III. MODÉLISATION

Notre objectif est qu'au lieu de considérer un seul MMC vertical ou horizontal pour modéliser une image donnée d'un mot manuscrit quelconque, nous construisons un RBD plus complexe en couplant deux MMCs. Ce couplage est réalisé en ajoutant des dépendances entre les variables d'états sous formes de liens (orientés) entre les nœuds du graphe. Dans ce cas, le problème rencontré est comment seront ajoutés ces liens.

A. Structure graphique du modèle RBD

La meilleure solution pour déterminer la structure graphique du modèle est d'apprendre cette structure à partir des données. Cette technique, appelée apprentissage structurel [4], n'est pas l'objectif de notre approche. Dans le cadre de notre travail, nous allons fixer une structure graphique de RBD pour toutes les images de mots.

Le choix de la structure doit se baser sur les critères suivants :

- Le modèle construit doit avoir un nombre raisonnable de paramètres pour que la complexité des calculs reste abordable.
- Aucune variable continue ne doit avoir de fils discret afin de pouvoir appliquer l'algorithme d'inférence exacte (arbre de jonction).
- Des liens entre les variables cachées des deux MMCs doivent exister afin de modéliser les dépendances.

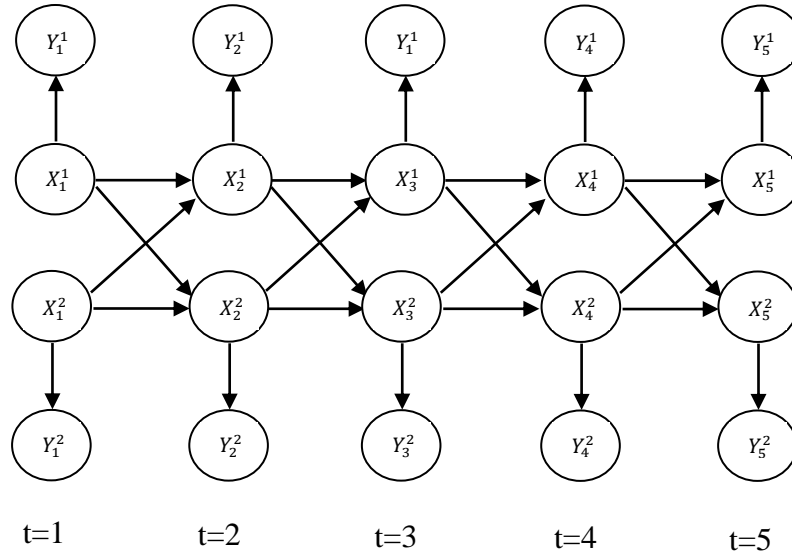


Figure 3. Modèle d'un RBD basé sur le couplage de 2 chaînes de Markov caché

Dans le modèle ci-dessus, les dépendances entre les deux MMCs modélisant les deux flux de données horizontal et vertical sont effectuées par des relations entre les états, conduisant à des modèles efficaces en termes de complexité du modèle [7] : un état d'un MMC est lié à l'état adjacent dans la tranche de temps suivant de l'autre MMC.

Nous supposons que le processus modélisé par le RBD est markovien de premier ordre et stationnaire. En pratique, cela signifie que les parents de toute variable X_t^i ou Y_t^i appartiennent seulement aux tranches de temps t et $t-1$ et que les paramètres du modèle sont indépendantes de t . Le modèle RBD peut être alors représenté par les deux premières tranches de temps. Il est entièrement décrit par la donnée des paramètres suivants :

$$\begin{cases} \pi = \{\pi_i^{(l)}\} = P(X_1^l = i), & l = 1, 2 \\ A \{a_{i,j,k}^{(l)}\} = P(X_t^l = k / X_{t-1}^1 = i, X_{t-1}^2 = j), & l = 1, 2 \text{ et } t \geq 2 \\ B = \{b_{j,k}^{(l)}\} = P(Y_t^l = k / X_t^l = j), & l = 1, 2 \text{ et } t \geq 2 \end{cases}$$

B. Prétraitement et extraction des caractéristiques de l'image d'un mot manuscrit

Chaque image d'un mot est transformée en deux séquences de vecteurs de caractéristiques qui seront les observations à passer au RBD: la première séquence de vecteurs de caractéristiques modélise le flux d'observations sur les colonnes et elle est calculée à partir d'une fenêtre glissante se déplaçant horizontalement de droite à gauche. La deuxième séquence de vecteurs de caractéristiques modélise le flux d'observations sur les lignes et elle est calculée à partir d'une fenêtre glissante se déplaçant verticalement de haut en bas (voir Figure 3.2). Dans notre cas, la fenêtre d'analyse est de taille uniforme. Aucune procédure de segmentation en caractère n'est faite à priori et chaque mot est transformé en deux matrices dont le nombre des lignes correspond au nombre de fenêtres et le nombre des colonnes correspond au nombre de caractéristiques.

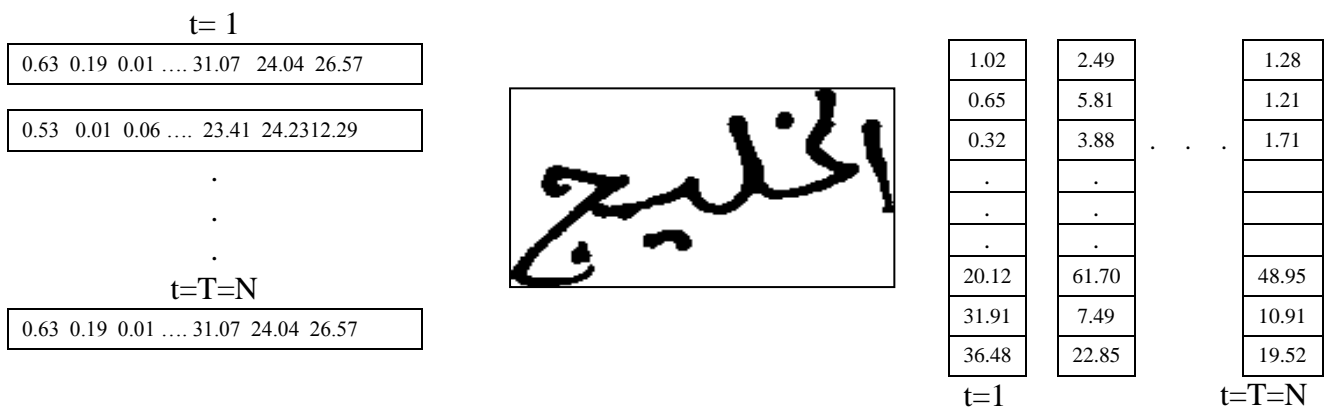


Figure 4. Exemple de descripteurs (7 moments de Hu et 6 moments de Zernike) calculés sur une image extraite de la base IFN/ENIT

Les caractéristiques calculées sur chaque fenêtre d'analyse sont :

- Les moments de Hu (7 moments) [8]
- Les moments de Zernike pris jusqu'à l'ordre 3 (6 moments) [9]

Dans notre approche, nous avons fait l'hypothèse que les caractéristiques extraites en balayant (horizontalement ou verticalement) l'image d'un mot peuvent être modélisé par un HMM discret. Pour cela nous avons recours à l'algorithme k-means pour discrétiser les observations obtenues.

C. Algorithme EM pour l'apprentissage de paramètres

Dans les expériences que nous avons effectuées, nous avons conçu un modèle par classe. Les modèles de toutes les classes partagent une architecture RBD unique, mais leurs paramètres diffèrent d'une classe à une autre. L'apprentissage des paramètres des modèles est effectué de façon indépendante modèle par modèle, en utilisant l'algorithme EM (Expectation Maximisation) [10].

L'algorithme EM est une approche itérative des estimateurs du maximum de vraisemblance. Chaque itération de l'algorithme EM est composée de deux étapes principales: une étape d'estimation (E) et une étape de maximisation (M). Le but de cet algorithme est de maximiser la fonction log de vraisemblance $l(\theta, Y) = \log(L(\theta, Y))$ où θ représente l'ensemble des paramètres du modèle et Y représente l'ensemble des observations relatives aux données du problème.

Pour appliquer l'algorithme EM pour l'apprentissage de notre modèle RBD, on doit noter qu'il s'agit d'un cas de données incomplètes. Dans ce cas, puisque l'ensemble des variables X est caché, $l(\theta, X, Y)$ devient une variable aléatoire et ne peut pas être maximisée directement. Donc l'algorithme EM s'appuie sur la maximisation de l'espérance sur la distribution de X , de la fonction "objectif", étant données les observations Y et les paramètres courants θ , en utilisant la fonction auxiliaire

$$Q(\theta, \hat{\theta}) = E_X[\log(L(\theta, X, Y)) / Y, \hat{\theta}]$$

1) Dans l'étape (E) d'estimation, on part de l'ensemble des paramètres θ et on calcule la fonction auxiliaire

$$Q(\theta, \hat{\theta}) = E_X[\log(L(\theta, X, Y)) / Y, \hat{\theta}]$$

- Dans le cas de variables cachées continues :

$$Q(\theta, \hat{\theta}) = \int_{x \in X} \log(L(\theta, x, y)) \times L(x/y, \hat{\theta}) dx$$

- Dans le cas de variables cachées discrètes :

$$Q(\theta, \hat{\theta}) = \sum_{x \in X} \frac{L(x, y / \hat{\theta})}{L(y / \hat{\theta})} \times \log(L(x, y / \theta))$$

Le rôle de l'espérance à postériori calculée dans la fonction précédente est de résoudre le problème de la présence des variables cachées variables X

2) Dans l'étape (M) de maximisation, on désire déterminer les paramètres $\hat{\theta}_{new}$:

$$\hat{\theta}_{new} = \arg \max_{\theta} Q(\theta; \hat{\theta})$$

Cette étape permet de maximiser la fonction auxiliaire calculée dans l'étape (E) par rapport à l'ensemble des paramètres.

D. Reconnaissance

La reconnaissance peut être effectuée de deux façons différentes :

- Soit par la recherche du chemin optimal qui donnera la classe dans le cas d'un seul modèle pour toutes les classes ;
- Soit par la recherche du modèle discriminant dans le cas d'un modèle par classe.

1) Reconnaissance par recherche du chemin optimal

Lorsqu'on utilise un seul modèle pour toutes les classes, le problème de reconnaissance consiste à déterminer le chemin optimal correspondant à l'observation $O = O_1, \dots, O_T$ c'est-à-dire trouver la meilleure suite d'états $Q = Q_1 \dots, Q_T$ qui maximise $P(Q/O, \lambda)$. Cette suite est appelée la suite d'états de Viterbi. Pour résoudre ce problème, on peut utiliser l'algorithme de Viterbi.

2) Reconnaissance par recherche du modèle discriminant

Lorsqu'il s'agit d'une reconnaissance de formes (mots ou caractères manuscrits) faisant partie d'un vocabulaire limité et statique, on peut considérer un modèle pour chaque élément du vocabulaire (un modèle par classe). Le problème de reconnaissance revient donc à identifier le modèle qui produit la probabilité à posteriori la plus élevée étant donnée l'entrée (les observations). Il s'agit d'une reconnaissance de type MAP (Maximum A Posteriori) par le calcul des probabilités d'émission de la forme en utilisant les modèles que l'on combine avec les probabilités à priori des classes.

Critère MAP

Soient λ_i le i ème modèle et $O = O_1, O_2, \dots, O_T$ une suite d'observations, le problème de reconnaissance en utilisant le critère MAP peut être formulé comme suit :

$$\begin{aligned} \lambda &= \arg \max_i P(\lambda_i/O) \\ &= \arg \max_i \frac{P(\lambda_i) P(O/\lambda_i)}{P(O)} \\ &= \arg \max_i P(\lambda_i) P(O/\lambda_i) \end{aligned}$$

En supposant que la distribution à priori sur les modèles est uniforme, c'est-à-dire toutes les classes ont la même probabilité d'apparition, on peut écrire :

$$\begin{aligned} \lambda &= \arg \max_i P(\lambda_i/O) \\ &= \arg \max_i P(O/\lambda_i) \end{aligned}$$

Dans ce cas le critère MAP est équivalent au critère de maximum de vraisemblance. En général, on utilise plutôt le logarithme de vraisemblance (log-likelihood).

Calcul de $P(O/\lambda)$

Soit $O = O_1, \dots, O_T$ une suite d'observations et $Q = Q_1, \dots, Q_T$ la suite d'états associée. La probabilité d'observation de O est égale à la somme sur tous les chemins d'états possibles Q des probabilités conjointes de O et Q :

$$P(O/\lambda) = \sum_Q P(O, Q/\lambda)$$

Cette formule directe nécessite beaucoup de calculs (2^{2N^T} opérations). Une évaluation incrémentale ou inductive obtenue à l'aide de l'algorithme forward backward permet de réduire le coût de ces calculs. Nous présentons dans ce qui suit, le principe général de cet algorithme utilisé pour le calcul de $P(O/\lambda)$ dans notre modèle :

- On définit la variable forward α_t comme suit :

$$\alpha_t(j_1, j_2) = P(o_1, o_2, \dots, o_t, S_{t,j_1}, S_{t,j_2}/\lambda)$$

et la variable backward β_t comme suit:

$$\beta_t(j_1, j_2) = P(o_{t+1}, \dots, o_T / S_{t,j_1}, S_{t,j_2}, \lambda)$$

- On peut calculer de manière inductive ces deux variables comme suit :

$$\alpha_t(i_1, i_2) = \begin{cases} \prod_1 \pi_{j_1}^{(0)} \cdot b_{j_1}^{(0)}(o_1^1), & t = 1, l = 0, 1 \\ \sum_{j_1, j_2} \left(\alpha_{t-1}(j_1, j_2) \prod_1 b_{j_1}^{(0)}(o_t^1) \cdot \sum_{l'} a_{j_1 i_1'}^{(l', 1)} \right), & t > 1, l = 0, 1 \end{cases}$$

et

$$\beta_t(i_1, i_2) = \begin{cases} 1, & t = T \\ \sum_{j_1, j_2} \left(\beta_{t+1}(j_1, j_2) \prod_1 b_{j_1}^{(0)}(o_{t+1}^1) \cdot \sum_{l'} a_{i_1 j_1}^{(l', 1)} \right), & t < T \end{cases}$$

- La fonction de vraisemblance $P(O/\lambda)$ peut-être calculée comme suit :

$$P(O/\lambda) = \sum_{j_1, j_2, \dots, j_L} \alpha_T(j_1, j_2)$$

IV. RÉSULTATS EXPERIMENTAUX

Le modèle développé est mis au point en utilisant le toolbox BayesNet (BNT) avec matlab. Les expériences sont effectuées sur un ensemble de 9 classes (voir TAB.1) représentant des noms de ville/villages tunisiennes extraits de la base de données IFN/ENIT [11].

Table I. Noms des villes utilisés pour l'expérimentation du modèle

Classe	Nom de ville	Exemple d'images
C1	الرضاع	
C2	شعاع	
C3	نحال	
C4	مارث	
C5	شماخ	
C6	الخليج	
C7	الرقاب	
C8	الفايض	
C9	سيدي إبراهيم الزهار	

Pour chaque classe, nous avons constitué à partir de l'ensemble d'images correspondant dans la base

IFN/ENIT, trois parties pour l'apprentissage, la validation et les tests.

- Base d'apprentissage : 50% des images de chaque classe sont utilisés pour l'apprentissage des paramètres (Tables de Probabilités Conditionnelles CPT) du modèle
- Base de validation : 25 % des images de chaque classe sont utilisées pour régler différents paramètres du modèle
- Base de test : 25 % des images de chaque classe sont utilisées pour tester le modèle

Les premières expériences consistent à trouver, en utilisant la base de validation, le nombre d'états optimal pour le modèle de chaque classe. Les critères utilisés pour fixer ce nombre d'états sont le taux de reconnaissance et le cout d'inférence en termes de temps. Pour ce faire, il suffit de varier à chaque fois le nombre d'états et de déterminer le taux de reconnaissance et le cout d'inférence.

Pour les résultats qui suivent, l'apprentissage et l'inférence dans chacun des modèles sont effectués en tenant compte du nombre d'états optimal correspondant déterminé dans les expériences précédentes. Comme nous l'avons déjà dit, nous avons utilisé l'algorithme EM pour l'apprentissage des paramètres (tables des probabilités conditionnelles CPT) de chacun des modèles. Le nombre d'itérations de l'algorithme EM nécessaires pour effectuer l'apprentissage varie d'un modèle à un autre.

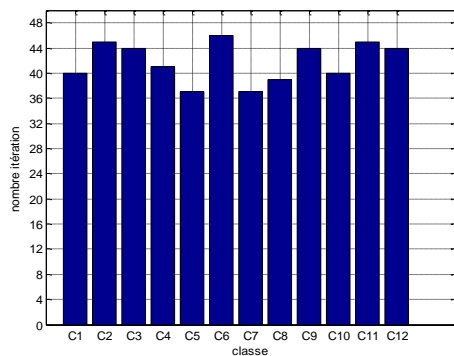


Figure 5. Variation du nombre d'itération (EM)

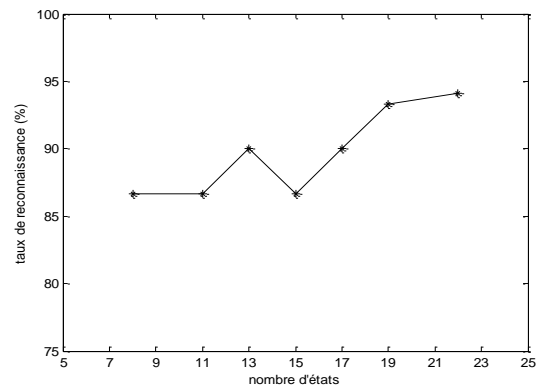


Figure 6. Courbe illustrant le taux de reconnaissance en fonction de nombre d'états pour la classe C2

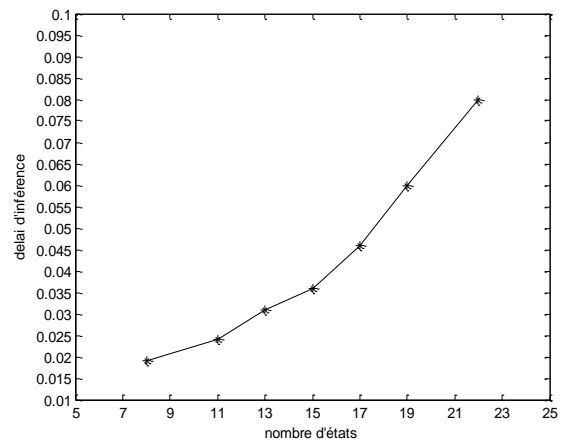


Figure 7. Courbe illustrant la durée d'inférence en fonction de nombre d'états pour la classe C2

Tous les résultats concernant le nombre optimal d'états pour chaque modèle sont illustrés dans le tableau suivant :

Table II. Nombre optimal d'états pour chaque classe

Classe	C1	C2	C3	C4	C5	C6	C7	C8	C9
Q	15	19	14	15	14	11	15	17	14

Après avoir trouvé le nombre optimal d'états pour chaque classe, nous avons effectués les tests de reconnaissance sur les échantillons de la base de test. Le tableau suivant illustre les taux de confusion générés par notre système pour la base de test (Table 3).

Les résultats utilisés dépendent des données utilisées pour l'apprentissage.

Table III. Taux de reconnaissance par classe

Classe	C1	C2	C3	C4	C5	C6	C7	C8	C9
Top1	75,8	72,2	67,9	70,4	70,9	76,2	75	74,7	77,4

Le taux moyen de reconnaissance T_m est défini comme étant la moyenne des taux T_i obtenus pour chaque classe de noms de villes :

$$T_m = \frac{\sum_{i=1}^n T_i}{n}, \text{ avec } n \text{ le nombre des classes considérées.}$$

Les taux moyen de reconnaissance obtenu sur la base de test est autour 73,5 %.

D'un autre coté, et comme l'illustre la matrice de confusion suivantes pour les 9 premières classes les réponses correcte figure toujours parmi les 4 premières réponses.

Table IV. Matrice de confusion

	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	80	5			5	10			
C2		72	14		10	4			
C3		11	74		10	5			
C4	8			77	6			9	
C5		18	7		73		2		
C6	16					76		8	
C7						8	68	24	
C8	6					7	16	71	
C9									75

D'autre, les tests réalisés sur la base d'apprentissage sont très encourageants (table 5) ce qui veut dire lorsqu'on disposera de suffisamment de données pour la phase d'apprentissage on pourra espérer de meilleurs résultats pour la base de tests

Table V. Résultats de la base d'apprentissage

Classe	C1	C2	C3	C4	C5	C6	C7	C8	C9
Top 1	91	81	89.2	92	91	91.3	89.3	88.5	90.2

V. CONCLUSION

Dans ce document, nous avons présenté un modèle de reconnaissance de l'écriture manuscrite basé sur les réseaux bayésiens dynamiques. Ce modèle se situe parmi les systèmes de reconnaissance, à vocabulaire limité, de l'écriture manuscrite hors ligne. Nous pensons que les résultats obtenus sont prometteurs et que beaucoup d'améliorations peuvent être apportées au modèle proposé, surtout en ce qui concerne les descripteurs calculés sur les blocs de l'image, la loi de distribution de ces descripteurs et les initialisations des algorithmes d'apprentissage.

REFERENCES

- [1] M. Deviren, K. Daoudi, Structural learning of dynamic Bayesian Networks in Speech Recognition, EUROSPEECH, 2001.
- [2] N. Ben Amara, A. Belaïd and N. Ellouze, Utilisation des modèles markoviens en reconnaissance de l'écriture arabe : État de l'art, CIFED, 2000.
- [3] K. Hallouli, L. Likforman-Sulem, M. Sigelle, A comparative study between decision fusion and data fusion in Markovian printed character recognition, in: Proceedings of ICPR, Quebec, 2002.
- [4] K. Hallouli, Reconnaissance de caractères par méthodes markoviennes et réseaux bayésiens dynamiques, Ph. D. Thesis pp. 89-97, ENST France, 2004 .
- [5] L. Likforman-Sulem, M. Sigelle, Recognition of degraded characters using dynamic Bayesian networks, Pattern Recognition, vol. 41, n° 10, pp. 3092-3103, TELECOM Paris, France, 2008
- [6] D. Bellot, Fusion de données avec les réseaux bayésiens pour la modélisation des systèmes dynamiques et son application en télémedecine, PhD thesis pp. 97-99, Université de Henri Poincaré – Nancy 1, 2002.
- [7] F. Slimane, Kamoun S., Hennebert J., Alimi M. A., Ingold R. Modèles de Markov Cachés et Modèle de Longueur pour la Reconnaissance de l'Écriture Arabe à Basse Résolution, Majec STIC, France 2009.
- [8] M.K. Hu. Pattern recognition by moment invariants. In Proc. IRE, Sept. 1961.
- [9] M.R. Teague. Image analysis via the general theory of moments. J. Optical Soc. Am., August 1980.
- [10] Y. Bengio and P. Frasconi. Input-Output HMMs for sequence processing. IEEE Trans. Neural Networks, 7(5):1231–1249, September 1996.
- [11] H. El Abed, V. Margner, Base de Données et Compétitions - Outils de Développement et d'Évaluation de Systèmes de Reconnaissance de Mots Manuscrits Arabes, CIFED, 2008 .
- [12] N. Ghanmy, MA. Mahjoub, NE Ben Amara, Characterization of FDynamic Bayesian Network, IJACSA, International Journal of Advanced Computer Science and Applications., vol 2, No 7, 2011