

Les Réseaux Bayésiens Dynamiques Hybrides

Appliqués à la Reconnaissance des Caractères Arabes En ligne

Redouane Tlemsani et Abdelhadi Lotfi

Laboratoire LaRATIC

Institut National des Télécommunication et des Technologies d'Information et de la Communication INTTIC 31000 ORAN ALGERIE

rtlemsani@ito.dz , ahd.lotfi@gmail.com

Abdelkader Benyettou

Laboratoire SIMPA

Université des Sciences et de la Technologie d'Oran mohamed Boudiaf USTO-MB 31000 ORAN ALGERIE

ak.benyettou@univ-usto.dz



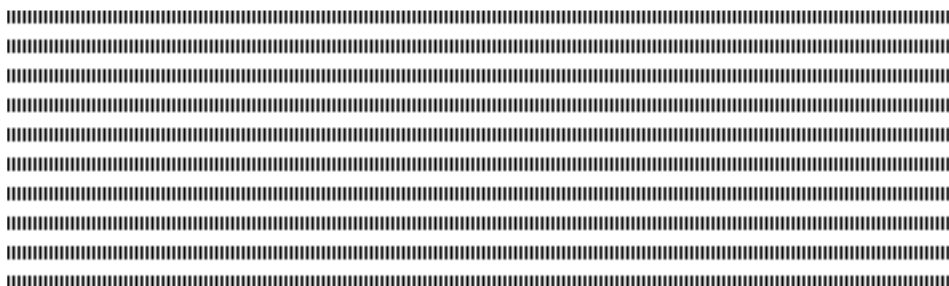
RÉSUMÉ. Notre système est un système Markovien que l'on peut voir comme un Réseau Bayésien Dynamique (Dynamic Bayesian Network DBN). L'approche est basée sur le formalisme des réseaux bayésiens dynamiques. La théorie des DBNs est une généralisation des réseaux Bayésiens aux processus dynamiques. L'application a porté sur la reconnaissance en ligne de caractères arabes isolés en utilisant notre base de données de laboratoire NOUN. Un testeur neuro-génétique a été proposé comme amélioration externe du DBN. Les scores pour DBN et DBN hybridé sont respectivement 66,78% et 73,47% ce qui laisse présager leur exploitation ultérieure ; d'autres approches d'amélioration tenant compte du temps ont été considérées et implémentées jusqu'à obtention d'un taux important 94,79% à base de reconnaissance.

ABSTRACT. This system is a Markovian system which one can see as like a Dynamic Bayesian Network (DBN). One of the major interests of these systems resides in the complete models training (topology and parameters) starting from training data. Our approach is based on the dynamic Bayesian Networks formalism. The DBNs theory is a Bayesians networks generalization to the dynamic processes. Our application will relate to the Arabic isolated characters on line recognition using our laboratory data base: NOUN. A neural tester proposed for DBN external optimization. The DBN scores and DBN mixed are respectively 70,24% and 62,50%, which lets predict their further development; other approaches taking account time were considered and implemented until obtaining a significant recognition rate 94,79%.

MOTS-CLÉS : Reconnaissance de Caractères Arabes en ligne, Réseaux Bayésiens Dynamiques

KEYWORDS: On line Arabic Character Recognition, Dynamic Bayesian Network.

CARI 2012



1. Introduction

Nous sommes intéressés ici dans la conception d'un système de reconnaissance capable d'apprendre et interpréter efficacement les caractères isolés et de s'adapter très rapidement au besoin spécifique d'utilisateur. Notre motivation principale dans ce domaine est d'étudier le manuscrit arabe via à vis de la technologie en ligne, autrement dit, le signal d'écriture en ligne acquis à partir d'une tablette électronique à digitaliser surtout que ceux d'applications d'écriture arabe en ligne ont vu le jour. Nous abordons aussi le problème de classification et le choix d'un modèle d'apprentissage pour ce type de système car les échantillons ont une forme spatiale, un sens du tracé et donc le choix de l'approche qui modélise mieux la trajectoire n'est pas toujours facile.

Dans ce domaine, nous proposons une nouvelle approche probabiliste : les réseaux bayésiens pour leur capacité d'effectuer des inférences dans un contexte d'incertitude, en quelque sorte comme alternative aux systèmes experts. On va les utiliser aussi pour leurs algorithmes d'apprentissage, comme alternative aux autres méthodes de modélisation quantitative, en les considérant comme des modèles de régression.

2. Réseaux Bayésiens Dynamiques

2.1. Principe

Les réseaux bayésiens dynamiques (DBN) prolongent la représentation de BN aux processus dynamiques. Un DBN code la probabilité jointe de distribution (JPD) d'un ensemble d'évolution de temps $X[t]=\{X_1[t], \dots, X_n[t]\}$ de variables. La JPD pour un intervalle fini de temps $[1, T]$ est factorisé comme :

$$p(X[1], \dots, X[T]) = \prod_{t=1}^T \prod_{i=1}^n P(X_i[t] | \Pi_i[t]) \quad (1)$$

Où $\Pi_i[t]$ dénotent les parents de $X_i[t]$ dans le graphique structurent du DBN. La structure graphique d'un DBN peut être regardée comme concaténation des plusieurs BNs statique lié avec les arcs temporels. Nous appelons chacun de ces réseaux statiques une tranche de temps du DBN. [1]

2.2. Représentation

La représentation de DBN est employée pour les 1^{iers} processus stationnaires. Pour ce cas, Friedman et autres décrit une représentation simplifiée en termes de deux charge statique BNs définie au-dessus des variables d'une tranche de temps [2]. Le réseau initial code la structure irrégulière dans la frontière et indique les états initiaux d'excédent de distribution $X[1]$. Le réseau de transition code la probabilité de transition $P(X[t+1] | X[t])$. Le JPD pour un intervalle fini de temps est obtenu en déroulant le réseau de transition pour un nombre suffisant de tranches de temps. Réarrangeant les limites dans Eq.(1), le JPD est factorisé des réseaux comme :

$$p(X[1], \dots, X[T]) = P_{B_1}(X[1]) \prod_{t=2}^{T-1} P_{B_t \rightarrow}(X_i[t] | X[t-1]) \quad (2)$$

Où $P_{B_i}(\cdot)$ et $P_{B_s}(\cdot)$ sont les densités de probabilité codées par les réseaux d'initiale et de transition, respectivement.

2.3. Inférence dans DBNs

L'idée est définis sur une couverture des variables qui emploie un procédé vers la procédure *forward-backward* du DBN [2,3,4]. Ces algorithmes sont une généralisation de l'algorithme (*Baum-Welch*) connu [5] dans les HMMs et cas spéciaux de l'algorithme de JLO [6].

2.4 Modèle Statique

Un exemple de stroke se compose de points. La figure 1 montre l'exemple récursif de construction de stroke. À la première itération récursive ($d=1$), IP_1 est ajouté pour modéliser des points médians de tous les exemples de stroke. Elle a les arcs d' EP_0 et d' EP_1 à IP_1 . À la deuxième itération récursive ($d=2$), IP_2 et IP_3 sont ajoutés pour des points médians des strokes partielles gauches et droites, respectivement. Figure.1 (c) est le modèle prolongé de stroke.

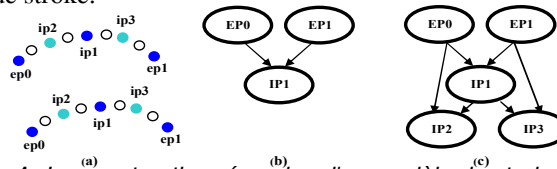


Figure 1. La construction récursive d'un modèle de stroke.

Avec ce procédé récursif, un modèle de stroke peut avoir autant de modèles de point selon les besoins. La probabilité jointe est calculée comme suit:

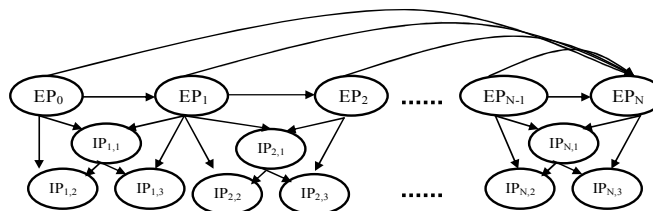
$$P(S=O(1), \dots, O(t)) = P\left(\begin{matrix} EP_0=O(1), EP_1=O(t), \\ IP_1=ip_1, \dots, IP_{t-1}=ip_{t-1} \end{matrix}\right) = P(EP_0=O(1))P(EP_1=O(t)) \times \prod_{i=1}^{t-1} P(IP_i=ip_i \setminus pa(IP_i)) \quad (3)$$

Où le $pa(IP_i)$ est la configuration des nœuds parents.

2.5 Modèle Dynamique

Un exemple de caractère se compose des strokes. Dans La Figure.2, EP_0 est le premier modèle de point écrit dans un caractère. Les modèles de point de premier stroke sont écrits dans l'ordre d' $IP_{1,2}$, $IP_{1,1}$, $IP_{1,3}$. Puis, les modèles de point des deuxièmes strokes sont écrits dans l'ordre d' EP_1 , $IP_{2,2}$, $IP_{2,1}$, $IP_{2,3}$. Alternativement, les strokes suivants sont écrits de la même façon. En conclusion, EP_N est le dernier modèle de point écrit dans un caractère [7,8,9].

Figure 2. La représentation par DBN d'un modèle de caractère avec N strokes et $d=2$.



Le modèle de vraisemblance d'un caractère est calculé par l'énumération de toutes les segmentations possibles de stroke. On pose un exemple de segmentation de stroke par $\gamma = (t_0, t_1, \dots, t_N)$, $t_0 = I < t_1 < \dots < t_N = T$, et l'ensemble de totalité par Γ . Ce modèle est donné comme suit :

$$\begin{aligned} P(O(1), \dots, O(t) \setminus BN) &= \sum_{\gamma=(t_0, \dots, t_N) \in \Gamma} P(S_1 = O(t_0, t_1), \dots, S_N = O(t_{N-1}, t_N)) \\ &= \sum_{\gamma \in \Gamma} \prod_{i=1}^N P(S_i = O(t_{i-1}, t_i) \setminus S_1 = O(t_0, t_1), \dots, S_{i-1} = O(t_{i-2}, t_{i-1})) \\ &= \sum_{\gamma \in \Gamma} \prod_{i=1}^N P(S_i = O(t_{i-1}, t_i) \setminus EP_0 = O(t_0), \dots, EP_{i-1} = O(t_{i-1})) \end{aligned} \quad (4)$$

Où $O(t_i, t_j) = O(t_i), O(t_{i+1}), \dots, O(t_j)$. La probabilité jointe donnée par des strokes précédents est calculée comme suit:

$$\begin{aligned} P(S_i = O(t_{i-1}, t_i) \setminus EP_0 = O(t_0), \dots, EP_{i-1} = O(t_{i-1})) &= \\ &\begin{cases} P(EP_i = O(t_i) \setminus O(t_0), \dots, O(t_{i-1})) \\ \prod_{j=1}^{i-1} P(IP_{i,j} = ip_{i,j}(O(t_{i-1}, t_i)) \setminus pa(IP_{i,j})) \text{ if } i > 1, \\ P(EP_0 = O(t_0)) P(EP_1 = O(t_1) \setminus O(t_0)) \\ \prod_{j=1}^{i-1} P(IP_{i,j} = ip_{i,j}(O(t_{i-1}, t_i)) \setminus pa(IP_{i,j})) \text{ if } i = 1, \end{cases} \end{aligned} \quad (5)$$

Où $ip_{i,j}(O(t_{i-1}, t_i))$ est le $j^{\text{ième}}$ point échantillon de $O(t_{i-1}, t_i)$. En substituant Eq. (5) pour Eq. (6), la probabilité du modèle ce n'est qu'un produit des probabilités jointes d'EPs et d'IPS :

$$P(O(1), \dots, O(t) \setminus BN) = \sum_{\gamma \in \Gamma} \prod_{i=0}^N P(EP_i = O(t_i) \setminus O(t_0), \dots, O(t_{i-1})) \times \prod_{i=1}^N \prod_{j=1}^{i-1} P(IP_{i,j} = ip_{i,j}(O(t_{i-1}, t_i)) \setminus pa(IP_{i,j})) \quad (6)$$

Les probabilités jointes des EPs peuvent être interprétées par des probabilités des positions de stroke globales et ceux des IPs avec des probabilités des strokes locales.

3. Reconnaissance et apprentissage

3.1 Algorithme de Reconnaissance

Un caractère est identifié en trouvant le modèle de caractère qui produit la probabilité a posteriori la plus élevée donnée. Quand le $i^{\text{ième}}$ modèle de caractère est dénoté BN_i et les points d'entrée en tant que $O(1), \dots, O(T)$, alors le problème de reconnaissance peut être formulé comme suit :

$$\begin{aligned} \operatorname{argmax}_i P(BN_i \setminus O(1), \dots, O(T)) &= \operatorname{argmax}_i \frac{P(BN_i) P(O(1), \dots, O(T) \setminus BN_i)}{P(O(1), \dots, O(T))} \\ &= \operatorname{argmax}_i P(BN_i) P(O(1), \dots, O(T) \setminus BN_i) \end{aligned} \quad (7)$$

Pour le calculer, toutes les segmentations possibles de stroke Γ sont considérées. Pour empêcher la complexité exponentielle de temps, nous supposons qu'elle peut être rapprochée par la probabilité jointe de caractère de la segmentation la plus probable γ^* dans Γ comme suit :

$$P(O(1), \dots, O(T) \setminus BN_i) \approx \max_{\gamma \in \Gamma} P(S_1 = O(t_0, t_1), \dots, S_N = O(t_{N-1}, t_N)) \quad (8)$$

Pour effectuer le calcul de la vraisemblance du modèle maniable dans le temps, nous avons besoin de plus une prétention pour la recherche de γ^* . Ensuite, l'algorithme peut être adopté comme suit :[10,11,12]

```

1.  $S_i$  :  $i^{\text{ème}}$  modèle de stroke
2.  $\gamma_i(t)$  : la segmentation la plus probable quand  $S_1, \dots, S_i$  et  $O(1,t)$  sont matchés.
3.  $\delta_i(t)$  : la probabilité jointe donnée  $\gamma_i(t)$ .
Initialisation
4.  $\delta_0(1)=1, \gamma_0(1)=\{\}$ 
Stroke matching
5. for  $t=2$  to  $T$ 
6. for  $i=1$  to  $N$ 
7.  $\delta_i(t) = \max_{1 \leq b < t} P(S_i = O(b,t) \setminus \gamma_{i-1}(b)) \cdot \delta_{i-1}(b)$ 
8.  $b^* = \arg \max_{1 \leq b < t} P(S_i = O(b,t) \setminus \gamma_{i-1}(b)) \cdot \delta_{i-1}(b)$ 
9.  $\gamma_i(t) = \gamma_{i-1}(b^*) \cup \{t\}$ 
10. end
11. end
Vraisemblance du modèle de caractère
12.  $P(O(1), \dots, O(T) | BN_i) \approx \delta_i(T)$ 
    
```

Figure 3. L'algorithme de programmation dynamique de recherche.

3.2 Algorithme d'Apprentissage

Les paramètres conditionnels de probabilité sont formés par des données d'apprentissage. Ils sont les matrices de régression linéaire W 's ($W=[w_{i,j}]$) et les covariances Σ 's pour les modèles de points (Eq 6) et (7)). Si tous les modèles de point sont assortis aux exemples de point, alors ils peuvent être estimés à partir des algorithmes statistiques conventionnels de régression avec l'objet de maximum de vraisemblance ML «*maximum likelihood*» [13]. Supposons qu'un point P dépend de P_1, \dots, P_k et il y a N échantillons d'apprentissage. On note le $i^{\text{ème}}$ échantillon de P as $p^{(i)}$ et les valeurs de variable dépendante par $z^{(i)}=[x^{(i)}_1, y^{(i)}_1, \dots, x^{(i)}_k, y^{(i)}_k, 1]$. Puis, ils sont estimés comme suit: [13]

$$W = \left(\sum_{i=1}^N p^{(i)} (z^{(i)})^T \right) \left(\sum_{i=1}^N z^{(i)} (z^{(i)})^T \right)^{-1} \tag{9}$$

$$\Sigma = \frac{1}{N} \sum_{i=1}^N p^{(i)} (p^{(i)})^T - \frac{1}{N} W \sum_{i=1}^N z^{(i)} (z^{(i)})^T \tag{10}$$

La ré-estimation des paramètres et la segmentation en strokes la plus probable γ^* est répété alternativement. [11,12,13] L'algorithme détaillé est comme suit :

1. Initialiser le modèle de caractère avec les données initiales (une partie des exemples des strokes manuellement segmentés).
2. Chercher la segmentation la plus probable γ^* de la totalité des caractères d'apprentissage non segmentés en utilisant l'algorithme de recherché précédente.
3. Estimer les paramètres (W et Σ) sur les exemples partitionnés par les γ^* .
4. Répéter les étapes 2 et 3 jusqu'à que la somme des vraisemblances du modèle ne changera plus (stabilité).

5. Expériences, Résultats et Analyses

5.1 Expérience 1 : Modèle Bayésien Dynamique

Nous avons utilisés notre base de données NOUN qui a été développée pour initier la recherche et le développement des systèmes de reconnaissance du manuscrit arabe en ligne. Elle contient 2800 caractères isolés, en arabe. Pour nos expériences, on a implémenté et vérifié notre modèle bayésien sur un ensemble de lettres de l'alphabet arabe. La figure 8 montre des taux relativement importants sur le sous corpus. Le taux global de reconnaissance atteint 66.78%.

5.2 Expérience 2 : Vérifieur Neuro-Génétique

Dans notre système, après avoir obtenu γ^* le vecteur des vraisemblances par l'algorithme de programmation dynamique, au lieu de prendre la valeur maximale, les chercheurs en reconnaissance des formes utilisent souvent un vérifieur à ce niveau pour adapter le système. L'architecture neuronale utilisée comme outil de vérification des formes, est montrée dans la figure ci-dessous.

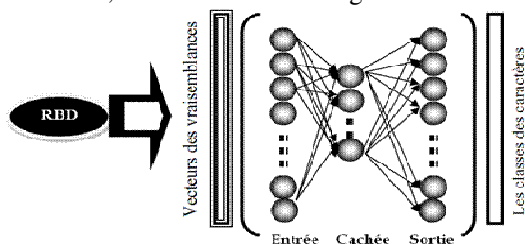


Figure 4. Vérifieur neuronale.

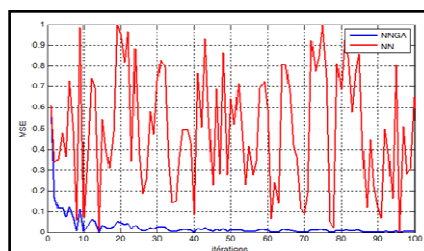


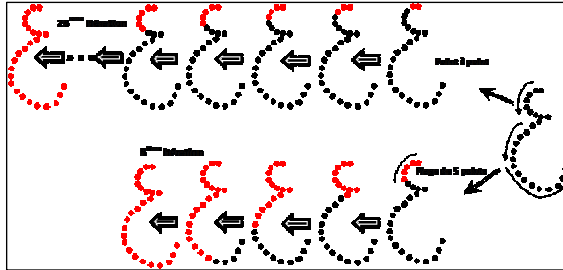
Figure 5. Vérifieur neuronale.

Dans la figure 5, on remarque la convergence de l'erreur quadratique moyenne obtenue à partir d'un apprentissage neuronale. Par la suite, une amélioration génétique a été appliquée sur les poids synaptiques des neurones associés et sur le même graphe, la convergence a été meilleure sur ce vérifieur. La figure 7 montre les taux de reconnaissance pour le modèle bayésien avec un vérifieur neuro-génétique (NNGA) et on a obtenu un taux plus élevé 73,47% c'est-à-dire une augmentation de 6,69%.

5.3 Expérience 3 : Time Segmentation « plage de points »

L'idée est de partager la trajectoire en des tranches de points, alors que cet algorithme donne une segmentation au niveau de chaque tranche de points, l'ancien algorithme consiste à affecter au niveau de chaque point du signal, une segmentation de ce dernier, ce qui est très lent pour un système de reconnaissance en ligne. On fixe ici un pourcentage de points au lieu d'un nombre fixe, ainsi le nombre de points des plages se change selon le nombre T total des points. L'idée de ce nouveau concept se résume dans la figure 6.

Figure 6. Algorithme de recherche dynamique (ARD) par plage de points



5.4 Discussions et comparaison

Dans un lieu de comparaison, la figure 7 a bien montré la progression du système d'une amélioration à une autre en fixant la profondeur du réseau à $d=3$.

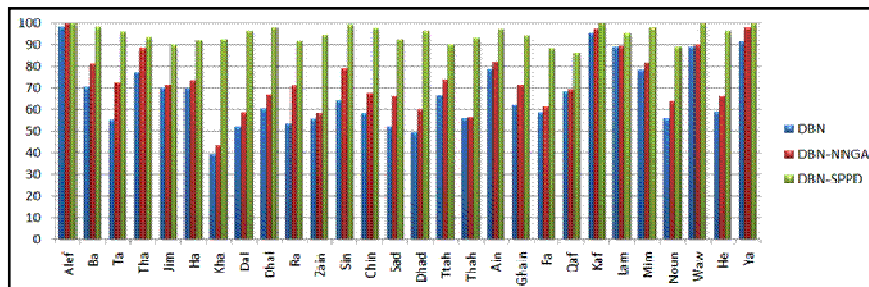


Figure 7. Scores en utilisant les trois approches d'amélioration implémentées

Les scores obtenus dans la première expérience débute par 39,03% pour le caractère Kha et passe pour le Qaf en moyenne en 68,32% et atteint le taux de reconnaissance maximum 98,33% pour le Alef. Nous avons pensé à une démarche hybride qui apparut efficace pour atteindre la fiabilité voulu au niveau de notre système. Et elle est affichée avec des petites améliorations des taux par exemple : les caractères Jim, Fa et Waw donnent respectivement du DBN à DBN-NNGA des faibles augmentations 1,41%, 2,60% et 1,02%. Cette insuffisance a poussée la recherche vers le mécanisme interne de ce réseau bayésien. Une amélioration du DBN-NNGA au DBN-SPPD atteint 49,10% pour le Kha et moyenne par 21,32%. Cette optimisation a fournie des meilleurs résultats par rapport le modèle classique. A la différence des travaux de [14], le taux de reconnaissance est égal à 95.17% (taux plus élevé que le notre qui égal à 94,79%) mais cela se justifie qu'ils ont utilisés les classes qui présentent la même forme géométrique. Dans notre système, les caractères ne sont pas regroupés.

6. Conclusion

Etudier une problématique de reconnaissance du manuscrit arabe est un défi particulier surtout via la technologie en-ligne sachant que peu de travaux commercialisés dans ce sens ont vu le jour. L'utilisation des modèles graphiques de connaissances, nous a permis de traiter efficacement l'ensemble des caractères isolés en

gardant leurs informations spatiales et l'ordre d'écriture de chaque caractère du manuscrit arabe. On a été confronté d'une part au traitement automatique de l'écriture arabe et en traitement en ligne et d'autre part au peu d'articles traitant l'écriture arabe en ligne.

7. Bibliographie

- [1] M. Deviren. Dynamic Bayesian networks for speech recognition. In Proceedings of AAAI 2002, SIGART/AAAI Doctoral Consortium, Edmonton, Canada, 2002.
- [2] G. Zweig. Speech Recognition with Dynamic Bayesian Networks. PhD thesis, University of California, Berkeley, Spring 1998.
- [3] N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. In UAI'98, Madison, Wisconsin, 1998.
- [4] K.P. Murphy. Dynamic Bayesian Networks: Representation, Inference and Learning. PhD thesis, UC Berkeley, Computer Science Division, 2002.
- [5] U. Kjærulff. A computational scheme for reasoning in dynamic probabilistic networks. In Proc of 8th Conference on Uncertainty in AI, pp 121-129, San Mateo, 1992.
- [6] L.R. Rabiner and R.W. Schafer. Digital Processing of Speech Signals. Prentice Hall, Englewood Cliffs, NJ, USA, 1978.
- [7] F.V. Jensen, S.L. Lauritzen, and K.G. Olesen. Bayesian updating in recursive graphical models by local computations. Computational Statistics and Data Analysis, 1990.
- [8] K. Nathan, H. Beigi, J. Subrahmonia, G.J. Clary, H. Maruyama, Real-time online unconstrained handwriting recognition using statistical methods, Proceedings of IEEE ICASSP, Detroit, USA, Vol. 4, 1995, pp. 2619-2622.
- [9] T. Starner, J. Makhoul, R. Schwartz, G. Chou, On-line cursive handwriting recognition using speech recognition methods, Proceedings of IEEE ICASSP, Adelaide, Australia, Vol. 5, 1994, pp. 125-128.
- [10] B.-K. Sin, J. Kim, Ligature modeling for online cursive script recognition, IEEE Trans. Pattern Anal. Mach. Intell. 19 (6) (1997) 623-633.
- [11] H. Kim, J. Kim, Hierarchical random graph representation of handwritten characters and its application to Hangul recognition, Pattern Recognition 34 (2) (2001) 187-201.
- [12] C.-L. Liu, I. Kim, J. Kim, Model-based stroke extraction and matching for handwritten Chinese character recognition, Pattern Recognition 34 (12) (2001) 2339-2352.
- [13] I.-J. Kim, J. Kim, Statistical utilization of structural neighborhood information for oriental character recognition, Proceedings of the Fourth IAPR International Workshop on Document Analysis Systems, Brazil, 2000, pp. 303-312.
- [14] N. Mezghani, A. Mitiche and M. Cheriet, A New Representation of Shape and its Use for Superior Performance in On Line Arabic Character Recognition by an Associative Memory, Int. J. on DAR, Vol. 7, N. 4, pp. 201-210, 2005.