Apprentissage de structures de réseaux bayésiens dynamiques pour la reconnaissance de la parole

Murat Deviren, Khalid Daoudi

LORIA/INRIA Équipe Parole
615 rue du jardin botanique 54602 Villers-lès-Nancy FRANCE
Mél: deviren,daoudi@loria.fr

RÉSUMÉ

We present a speech modeling methodology where no a priori assumption is made on the dependencies between the observed and the hidden speech processes. Rather, dependencies are learned from data. This methodology guarantees improvement in modeling fidelity as compared to HMMs. In addition, it gives the user a control on the trade-off between modeling accuracy and model complexity. We evaluate the performance of the proposed methodology in a connected digit recognition task.

1. INTRODUCTION

La grande majorité des systèmes actuels de reconnaissance automatique de la parole utilise une modélisation probabiliste du signal de parole par des modèles de Markov cachés (ou HMM pour Hidden Markov Models). Dans un HMM, les observations sont supposées être gouvernées par un processus dynamique caché. Les hypothèses d'indépendance associées sont telles que le processus caché est markovien de premier ordre et que chaque observation dépend seulement de la variable cachée actuelle. Il y a cependant une question fondamentale concernant ces hypothèses de dépendance : sont-elles consistantes avec les données et avec tout type d'application?.

Dans [6], nous avons proposé une méthode dans laquelle nous ne faisons aucune hypothèse de dépendance *à priori*. Plutôt, nous donnons aux données une liberté complète (mais contrôlée) pour dicter les dépendances appropriées. En d'autres termes, nous apprenons les dépendances entre les variables (cachées et observables) à partir des données. Le principe de cette méthode est de rechercher toutes les dépendances "réalistes" possibles, et à choisir celles qui expliquent au mieux les données. Cette approche a l'avantage de garantir que le modèle résultant représente la parole avec une plus grande fidélité que les HMM. En outre, un contrôle est donné à l'utilisateur pour faire un compromis entre la fidélité et la complexité du modèle. De plus, l'approche est techniquement très attrayante parce que tout l'effort de calcul est fait dans la phase d'apprentissage.

Notre approche est basée sur le formalisme des réseaux Bayésiens dynamiques (ou DBNs pour Dynamic Bayesian Networks). La théorie des DBNs est une généralisation des réseaux Bayésiens (ou BNs Bayesian Networks) aux processus dynamiques. Brièvement, le formalisme des réseaux Bayésien consiste à associer un graphe acyclique orienté à une distribution jointe de probabilités (ou JPD pour Joint Probability Distribution) $P(X)$ d'un ensemble de variables aléatoires $X = \{X_1...,X_n\}$. Les noeuds de ce graphique représentent les variables aléatoires, et les flèches codent les indépendances conditionnelles (IC) qui sont supposées dans le JPD. L'ensemble de toutes les relations d'IC que les propriétés de séparation du graphe impliquent, se nomment les propriétés de Markov. Un BN est complètement défini par une structure de graphe S et un jeu de paramètres Θ de probabilités conditionnelles des variables étant donné leurs parents. En effet, le JPD peut être exprimée sous une forme factorisée qui est, $P(X) = \prod_{i=1}^{n} P(X_i|\Pi_i)$, où Π_i dénote les parents de X_i dans S.

L'utilisation de DBNs dans la reconnaissance de la parole a suscité beaucoup d'intérêt ces dernières années [1, 2, 12]. Dans cet article, nous utilisons la flexibilité de cette approche, et au lieu de fixer à priori la structure des modèles acoustiques (comme on ferait avec les HMM), nous établissons un système " intelligent " qui fonctionne comme suit. Nous alimentons le système en utilisant les données observées. Puis, le système détermine la structure (càd les dépendances) et les paramètres Θ qui représentent au mieux les données. Cette stratégie est connue comme l'apprentissage de structures dans la littérature des BNs. Dans [6], nous avons utilisé cette méthodologie d'apprentissage pour une tâche de reconnaissance de mots isolés. Nous étendons l'application de la méthodologie à la reconnaissance de mots connectés dans [5]. Dans cet article, nous présentons une vue d'ensemble de l'approche et nous l'évaluons à travers un problème de reconnaissance de chiffres connectés.

Dans la section suivante, nous présentons une brève introduction aux réseaux Bayésiens dynamiques. Dans la section 3, nous définissons la classe de DBNs que nous utilisons dans notre configuration. Ensuite, nous récapitulons brièvement l'algorithme de l'apprentissage de structures. Dans la section 5, nous décrivons l'algorithme de décodage pour la reconnaissance de la parole en utilisant les DBNs. Enfin, nous évaluons le potentiel de notre approche sur une tâche de reconnaissance de chiffres connectés.

2. RÉSEAUX BAYÉSIENS DYNAMIQUES

Un DBN code la distribution jointe de probabilités d'un ensemble de variables $X[t] = \{X_1[t], \ldots, X_n[t]\}$ évoluant dans le temps. Si nous considérons T pas de temps, le DBN peut être considéré comme un BN (statique) avec $T \times n$ variables. En utilisant la propriété de factorisation des BNs, la densité jointe de probabilités de $\mathbf{X}_1^T = \{X[1],$

$\ldots, X[T]\}$ est :

$$P(X[1],\ldots,X[T]) = \prod_{t=1}^{T}\prod_{i=1}^{n} P(X_i[t]|\Pi_{it}) \quad (1)$$

où Π_{it} dénote les parents de $X_i[t]$. Dans la littérature des BNs, les DBNs sont définis en faisant l'hypothèse que $X[t]$ est un processus markovien [8]. Dans cet article, nous affaiblissons cette hypothèse pour permettre des processus non-markovien et pour que le processus $X[t]$ satisfasse à :

$$P(X_i[t]|\mathbf{X}_1^{t+\tau_f}) = P(X_i[t]|X[t-\tau_p],\ldots,X[t+\tau_f]) \quad (2)$$

pour tous les nombres entiers positifs τ_p et τ_f. Graphiquement, l'hypothèse ci-dessus déclare qu'une variable au temps t peut avoir des parents dans l'intervalle $[t-\tau_p, t+\tau_f]$. Cependant, un soin particulier doit être pris avec les variables aux limites (voir le [6] pour les détails). Dans cette perspective, il est facile de représenter un HMM comme un DBN. En effet, les propriétés de Markov (les hypothèses de dépendance) d'un HMM, sont encodées par la structure graphique représentée sur le Figure 1. Chaque noeud de cette structure représente une variable aléatoire $X_h[t]$ ou $X_o[t]$, dont la valeur indique l'état ou l'observation au temps t.

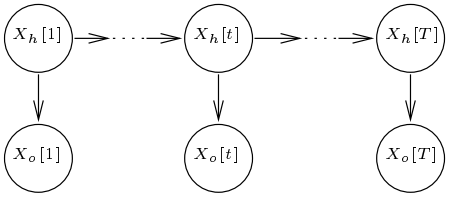

FIG. 1 – *HMM représentés comme DBN*

3. CLASSE DE RECHERCHE DE STRUCTURES

L'apprentissage structures de DBNs exige l'utilisation d'une classe de structures. La recherche à travers toutes les structures possibles de DBNs nécessite un temps de calcul exponentiel. Par conséquent, nous nous limitons à un petit mais riche ensemble de structures qui représente seulement des dépendances réalistes, d'un point de vue physique et algorithmique. Le lecteur peut se référer à [6] pour le raisonnement sur les dépendances autorisées qui sont définies comme suit. Soit $X[t] = \{X_h[t], X_o[t]\}$ l'ensemble de variables cachées et observées au temps t. Nous supposons que :

– la variable cachée au temps t est indépendante de $\mathbf{X}_1^{t-\kappa-1}$ sachant les κ dernières variables cachées, pour $t > \kappa$,

$$P(X_h[t]|\mathbf{X}_1^{t-1}) = P(X_h[t]|X_h[t-\kappa],\ldots,X_h[t-1]). \quad (3)$$

– la variable d'observation au temps t est indépendante de toutes les autres variables sachant les variables cachées dans l'intervalle $[t-\tau_p, t+\tau_f]$, pour tous les nombres entiers positifs τ_p et τ_f,

$$P(X_o[t]|\mathbf{X}_1^T \setminus \{X_o[t]\}) = P(X_o[t]|X_h[t-\tau_p], \ldots, X_h[t+\tau_f]). \quad (4)$$

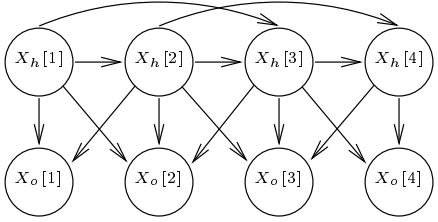

FIG. 2 – *Structure de DBN* $(\kappa,\tau_p,\tau_f) = (2,1,1)$, $T = 4$

Par conséquent, la classe de recherche des structures de DBN permises est définie par les triplets (κ,τ_p,τ_f) pour, $1 \le \kappa \le \kappa_{max}$, $0 \le \tau_p \le \tau_{p_{max}}$, $0 \le \tau_f \le \tau_{f_{max}}$, où $(\kappa_{max},\tau_{p_{max}},\tau_{f_{max}})$ est une limite supérieure à la taille de la classe de recherche. À la limite inférieure $(\kappa,\tau_p,\tau_f) = (1,0,0)$, la structure se réduit à un HMM standard d'ordre 1 (Figure 1) où, Eq.(3) définit les probabilités de transition d'état et l'Eq.(4) définit les probabilités d'observation. Chaque triplet (κ,τ_p,τ_f) spécifie une structure de DBN. La Figure 2 illustre le cas où $(\kappa,\tau_p,\tau_f) = (2,1,1)$.

Si chaque variable cachée discrète $X_h[t]$ prend ses valeurs dans l'ensemble ordonné d'étiquettes $I = \{1 \ldots N\}$, et chaque variable observable a une densité gaussienne conditionnelle, la paramétrisation numérique de nos DBNs est la suivante :

$$\begin{aligned} P_v(X_h[t] = j|\Pi_{ht} = \mathbf{i}) &= a_{\mathbf{i}j}[t], \, for \; j \in I \\ P_v(X_o[t]|\Pi_{ot} = \mathbf{i}) &\sim \mathcal{N}(\mu_{\mathbf{i}}[t],\Sigma_{\mathbf{i}}[t]). \end{aligned} \quad (5)$$

L'index $\mathbf{i}$ prend ses valeurs dans l'ensemble des valeurs possibles des variables parents. Pour la classe spécifique des structures que nous considérons, les parents sont toujours les variables cachées, ainsi, $\mathbf{i}$ est un point dans l'espace cartésien I^m, où m est le nombre de parents de la variable considérée.

Afin de fournir des comparaisons justes dans les expériences, nous calculons le nombre de paramètres requis pour définir le DBN. Soit M le nombre de paramètres requis pour encoder la densité de probabilité d'observation. Pour une topologie de gauche à droite, le nombre exigé de paramètres pour encoder le réseau est donné par:

$$C = 2^\kappa N - \sum_{l=1}^{\kappa} l2^{\kappa-l} + M \times [2^\beta N - \sum_{l=1}^{\beta} l2^{\beta-l}]. \quad (6)$$

β est défini par $\beta = \tau_p + \tau_f$ (nous supposons que $\kappa,\beta \le N$).

4. APPRENTISSAGE DE STRUCTURES

Dans le formalisme des BNs, le problème de l'apprentissage a deux aspects : l'apprentissage des paramètres sachant une structure et l'apprentissage de la structure. Dans notre situation nous utilisons l'approche du maximum de vraisemblance (ou ML pour Maximum Likelihood) pour le premier cas. Étant donné un DBN avec la structure S et un ensemble d'observations, les paramètres Θ sont choisis tels que la probabilité des observations soit maximisée. Ces paramètres peuvent être calculés en utilisant l'algorithme EM. Dans [4] nous dérivons les équations de mise à jour de EM pour la classe des DBNs que nous avons définie dans la section précédente.

L'apprentissage de la structure d'un BN est réalisée par une recherche dans un espace de structures possibles. La

taille de cet espace de recherche est un problème critique. Dans la section précédente nous avons défini un ensemble de structures réalistes et plausibles pour modéliser la parole. Dans ce qui suit, nous récapitulons le principe général de l'algorithme d'apprentissage de structures. Les détails de l'algorithme et les références sont dans [6]. Étant donné un ensemble d'observations, notre but est de trouver la structure optimale définie par le triplet (κ,τ_p,τ_f), et les probabilités conditionnelles associées qui expliquent au mieux les données. L'optimalité est atteinte quand la probabilité des observations est maximale et la complexité de la structure est minimale. Il y a fondamentalement deux méthodes d'évaluation du degré auquel une structure s'adapte aux données : la métrique bayésienne de Dirichlet (BD), et Minimum Description Length (MDL) (ou d'une manière équivalente le Bayesian Information Criteria (BIC)) [9]. Nous utilisons la métrique MDL, qui pénalise les structures complexes basées sur le nombre de paramètres employés pour coder le modèle. La métrique MDL est définie de la façon suivante :

$$Score_{MDL} = \log P(D|\Theta,S) - \frac{\log L}{2} \sum_{i=1}^{n} ||X_i,\Pi_i|| \quad (7)$$

où D est l'ensemble des observations, L est le nombre d'exemples (réalisations) dans D et $||X,Y||$ est défini comme le nombre de paramètres requis pour coder la probabilité conditionnelle, $P(X|Y)$. Le terme de vraisemblance peut être calculé en utilisant l'algorithme de JLO qui est un algorithme efficace d'inférence pour les réseaux Bayésiens [10].

Afin de trouver le modèle optimal, nous employons l'algorithme EM structurel (SEM pour Structural Expectation Maximization) [7]. L'algorithme commence avec une structure quelconque et des paramètres initiaux. À chaque étape, les scores des structures candidates sont calculés avec la métrique MDL. La structure qui obtient le score maximal est choisie comme prochaine structure, et les paramètres de cette structure sont mis à jour avec une étape de EM paramétrique. En mettant à jour la structure itérativement à chaque étape, l'algorithme SEM garantir une augmentation des scores et une convergence vers un maximum local [7].

Dans notre application, nous initialisons l'algorithme avec $(\kappa_{max},\tau_{p_{max}},\tau_{f_{max}})$ comme la limite supérieure sur l'espace de recherche de structure et nous utilisons la structure de HMM pour la première itération. Cette initialisation garantit que le modèle résultant aura une fidélité plus élevée (ou égale), par rapport au HMM. Le compromis entre la complexité de l'algorithme d'apprentissage et la fidélité du modèle résultant est contrôlée par la limite supérieure sur l'espace de recherche.

5. ALGORITHME DE DÉCODAGE

Dans cette section, nous discutons brièvement l'algorithme de décodage dans des tâches de reconnaissance de mots isolés et connectés en utilisant la classe de DBNs que nous avons décrite. Soit un vocabulaire V de $|V|$ mots et un modèle de DBN pour chaque mot $v \in V$. L'algorithme d'apprentissage proposé nous permet de capturer les différentes structures de dépendance pour chaque mot. Par conséquent, les modèles appris ont (probablement) des structures $(\kappa^v,\tau_p^v,\tau_f^v)$ différentes pour chaque mot $v \in V$.

Considérons alors la tâche de reconnaissance de mots isolés. Sachant une séquence d'observation, le problème est de trouver le meilleur DBN tel que la probabilité des observations soit maximale. Dans les DBNs, la vraisemblance peut être calculée en utilisant l'algorithme JLO [10]. Cet algorithme est une version généralisée de l'algorithme Forward-Backward dans les HMMs [11]. Ainsi la classification est réalisée en opérant l'algorithme JLO sur chaque modèle de DBN et en choisissant celui qui donne le maximum de vraisemblance. Dans [6], nous présentons les résultats d'une tâche de reconnaissance de chiffres isolés en utilisant l'approche d'apprentissage décrite dans la section précédente.

La reconnaissance de mots connectés est une tâche plus compliquée. Le problème du décodage consisté à identifier la séquence la plus vraisemblable de mots, étant donnée une phrase prononcée. À chaque instant, la probabilité de chaque mot doit être calculée et le meilleur ordre des mots doit être choisi pour maximiser la probabilité globale. Dans les HMMs, ceci est fait en utilisant l'algorithme de Viterbi. Dans notre configuration il y a deux étapes canoniques pour faire le décodage: 1) un algorithme pour trouver la meilleur séquence d'états dans un DBN sachant une séquence d'observation, 2) une méthodologie pour "augmenter" le modèle DBN avec le modèle de langage. La première étape est effectuée grâce à l'algorithme de Dawid [3] ce qui peut être vu comme une version plus générale de l'algorithme de Viterbi. Pour la deuxième étape, nous proposons dans [5] une méthode pour construire un méta-modèle qui encapsule tous les DBNs de mots et le modèle de langage. Quand les DBNs des mots ont différentes structures, la construction du modèle augmenté a besoin d'une attention particulière. Nous construisons le modèle augmenté en utilisant une structure qui représente toutes les relations de dépendance dans tous les modèles de mots. Afin de respecter les différentes IC (Indépendances Conditionnelles) de chaque DBN, nous employons une paramétrisation spéciale où nous codons les IC dans les paramètres du DBN plutôt que dans la structure. Une fois qu'un tel DBN est construit, l'algorithme de Dawid peut être employé sur ce DBN pour trouver la meilleure séquence de mots. Les détails de cette procédure peuvent être trouvés dans [5].

6. EXPÉRIENCES

Dans cette section, nous comparons [1] les exécutions des modèles à base de DBNs et à base de HMMs standards. Nos expériences sont effectuées sur la base de données *Tidigits*. Dans l'apprentissage, nous utilisons seulement la partie contenant les mots isolés de la base de données d'apprentissage où chaque locuteur prononce 11 chiffres ('o', $0,1,\ldots 9$) deux fois. L'étape initiale a été de segmenter le corpus d'apprentissage aux régions de silence. Nous faisons un apprentissage isolé pour chaque modèle de HMM et de DBN utilisant cette première segmentation. Dans toutes les expériences, le silence est modélisé par un HMM à un état. Dans les expérimentations, nous

1. Des très bons scores peuvent être obtenus sur cette base de données en utilisant des mixtures de gaussiennes HMMs et des paramètres ajustés. Notre but ici n'est pas d'accorder les paramètres afin de réaliser les rendements les plus élevés. Nous voulons plutôt, fournir des comparaisons justes en utilisant des systèmes de base. Nous croyons que de cette façon, nous pouvons porter un jugement initial juste sur les capacités de chaque système.

utilisons toute la base de données (de test) dans laquelle 8636 phrases sont prononcées, chaque phrase contenant entre 1 et 7 chiffres. La taille de chaque vecteur d'observation est 35. Ils se composent de 11 MFCCs (énergie enlevée), de 12 Δ, et de 12 $\Delta\Delta$. La matrice de covariance est supposée diagonale. Nous utilisons une topologie de transition de gauche à droite et un modèle uniforme de langage, càd, $P(v|v') = \frac{1}{12}$ ($|V| = 12$). Dans l'algorithme d'apprentissage de structures, pour chaque chiffre, la limite supérieure de l'espace de recherche est $(\kappa_{max},\tau_{p_{max}},\tau_{f_{max}}) = (2,1,1)$.

(κ,τ_p,τ_f)	N=4	N=5	N=6
(1,0,0)			'o',0,2,7
(1,0,1)	'o',2,4,7,8,9	'o',1,4,5,9	3,6,8
(1,1,0)	0,1,3,5,6	0,2,3,6,7,8	1,4,5,9

TAB. 1 – *Résultats de l'algorithme SEM.*

Nous effectuons des expériences en utilisant un nombre variable d'états, càd différentes valeurs pour N. Dans le Tableau 1, nous montrons les résultats de l'algorithme d'apprentissage de structures. Quand $N \geq 7$, notre système montre que le HMM standard est le meilleur modèle pour tous les chiffres. Ceci prouve que quand un HMM est suffisant pour modéliser des données, notre système le reconnaît. Les taux de reconnaissance sont donnés dans le Tableau 2, pour $N = 4,5,6$. Notre système surpasse en grande partie le système à base de HMMs, en particulier quand $N = 4$. Cette augmentation remarquable de l'exécution peut être expliquée comme suit. Puisque le nombre d'états cachés est petit, le système à base de HMMs conduit à beaucoup d'insertions. En utilisant notre système, tous les modèles appris impliquent des dépendances de contexte. Il est bien connu que la modélisation du contexte améliore la modélisation de durée. Par conséquent, quoique le nombre d'états soit encore petit, notre système fournit une meilleure modélisation de la durée des chiffres, et le nombre d'insertions est ainsi énormément réduit. Cependant, pour faire des comparaisons justes, nous devrions comparer notre système à un HMM, avec un nombre équivalent de paramètres. Pour ce faire, nous comparons notre système à un HMM avec le même nombre d'états, mais nous utilisons un mélange de 2 gaussiens comme densité de probabilité des observations. Le nombre moyen de paramètres C_{av} de chaque système est calculé en faisant la moyenne de la quantité C définie par Eq.(7) avec de tous les modèles de chiffre[2]. Pour $N = 4,5$, notre système ($C_{av} = 497, 639$) donne toujours de meilleurs résultats que le HMM_{2G} (qui utilise bien plus de paramètres que notre système $C_{av} = 567, 709$). Pour $N = 6$, quoique notre système utilise beaucoup moins de paramètres ($C_{av} = 653.6$) que HMM_{2G} ($C_{av} = 1551$), les deux systèmes donnent approximativement les mêmes résultats. Comme remarque finale sur ces expériences, nous

N	HMM(N)	DBN(N)	$\text{HMM}_{2G}(N)$
4	20.57	63.22	46.32
5	60.92	79.86	77.99
6	77.93	84.47	84.44

TAB. 2 – *Précision d'identification des chiffres (%).*

noterons que notre système permet l'apprentissage des processus non-Markoviens (le DBN (1,0,1) est non-Markovien, par exemple). Ceci est un avantage important par rapport aux HMMs. En effet, en faisant ainsi, notre système peut tenir compte du phénomène *d'anticipation* bien connu qui se produit dans le mécanisme de production de la parole. Ces expériences montrent la puissance de l'approche des DBNs pour modéliser les aspects acoustiques et phonétiques de la parole avec une fidélité plus élevée que les HMMs.

2. Pour 1 gaussien $M = 70$, pour 2 gaussiens $M = 140$.

7. CONCLUSION

Nous avons utilisé le formalisme des DBNs pour construire des modèles acoustiques qui sont capables d'apprendre la structure de dépendance entre le processus caché et observé de la parole. Nous avons présenté une méthodologie pratique pour utiliser de tels modèles dans la reconnaissance de la parole continue. L'approche permet l'utilisation de modèles à différents structures pour différentes mots dans le vocabulaire. Nous avons montré qu'en utilisant nos modèles, nous pouvons obtenir de meilleurs résultats de reconnaissance qu'avec des HMMs équivalents.

RÉFÉRENCES

[1] J. A. Bilmes. Dynamic bayesian multinets. In *UAI*, 2000.

[2] K. Daoudi, D.Fohr, and C. Antoine. Continuous multi-band speech recognition using bayesian networks. In *ASRU*, 2001.

[3] A.P. Dawid. Application of a general propagation algorithm for probabilistic expert systems. *Statistics and Computing*, (2):25–36, 1992.

[4] M. Deviren. Structural learning of dynamic Bayesian networks in speech recognition. Technical report, Loria, 2001.

[5] M. Deviren and K. Daoudi. Continuous speech recognition using structural learning of dynamic Bayesian. preprint, www.loria.fr/ deviren.

[6] M. Deviren and K. Daoudi. Structural learning of dynamic Bayesian networks in speech recognition. In *Proceedings of Eurospeech'2001*, Denmark, September 2001.

[7] N. Friedman. Learning belief networks in the presence of missing values and hidden variables. In *Int. Conf. Machine Learning*, 1997.

[8] N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. In *UAI*, 1998.

[9] D. Heckerman. A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Advanced Technology Division, March 1995.

[10] F.V. Jensen, S.L. Lauritzen, and K.G. Olesen. Bayesian updating in recursive graphical models by local computations. *Computational Statistics and Data Analysis*, 4:269–282, 1990.

[11] P. Smyth, D. Heckerman, and M. I. Jordan. Probabilistic independence networks for hidden markov probability models. *Neural Computation*, 9(2):227–269, 1997.

[12] G. Zweig. *Speech Recognition with Dynamic Bayesian Networks*. PhD thesis, University of California, Berkeley, Spring 1998.