Modèles de Markov Cachés

Joël Le Roux

14 mars 2003

Table des matières

1 Introduction

A l'heure actuelle, il n'existe guère d'applications commerciales de méthode efficace de description du signal vocal sur laquelle on peut baser la reconnaissance. La technique qui s'est avérée la meilleure pour le moment est la recherche d'une similitude entre des chaines de vecteurs mémorisées représentant par exemple des mots et celle qu'on déduit du signal enregistré. Nous verrons au chapitre 3 une présentation succinte de la mise en œuvre de cette technique.

Notons que les modèles de Markov cachés sont une approche prometteuse dans différents domaines d'applications où on envisage de traiter des données quantifiées qui peuvent être partiellement erronnées comme par exemple

- La reconnaissance d'images : caractères, empreintes digitales, ...
- la recherche de motifs et de séquences dans les gènes ; on trouvera plusieurs références sur le site

`http://www.cse.ucsc.edu/research/compbio/html_format_papers/hughkrogh96/cabios.html.`

2 Modèles de Markov cachés

La présentation de ce document s'appuie sur l'article de L. Rabiner "A tutorial on hidden markov models and selected applications in speech recognition", Proceedings of the IEEEE, vol. 77, no 2, Feb 1989.

2.1 Le modèle de production des données

Une chaine de Markov cachée est un automate à M états que nous noterons

$$1, \ldots, m, \ldots, M. \tag{1}$$

Nous noterons s_t l'état de l'automate à l'instant t.

La probabilité de transition d'un état m à un état m' est donnée ; nous l'appellerons $a(m, m')$.

$$a(m, m') = p(s_t = m'/s_{t-1} = m). \tag{2}$$

On a

$$\sum_{m'=1}^{M} a(m, m') = 1. \tag{3}$$

On se donne aussi $d(m)$ la probabilité que l'automate soit dans l'état m à l'instant initial :

$$d(m) = p(s_0 = m). \tag{4}$$

On a

$$\sum_{m=1}^{M} d(m) = 1. \tag{5}$$

Lorsque l'automate passe dans l'état m il émet une donnée y_t qui peut prendre N valeurs :

$$1, \ldots, n, \ldots, N. \tag{6}$$

La probabilité pour que l'automate émette un signal n lorsqu'il est dans l'état m sera notée $b(m, n)$:

$$b(m, n) = p(y_t = n/s_t = m). \tag{7}$$

On a

$$\sum_{n=1}^{N} b(m, n) = 1. \tag{8}$$

L'adjectif "caché" employé pour caractériser le modèle traduit le fait que l'émission d'une donnée à partir d'un état est aléatoire. C'est ce caractère aléatoire des mesures qui, ajouté aux propriétés des processus markoviens fait la souplesse et la puissance de l'approche proposée par F. Jelinek.

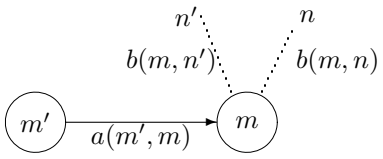

Figure 1: Probabilité de transition et d'émission d'une donnée dans le modèle de Markov caché

Ce dernier s'est appuyé sur la travail théorique de L. E. Baum et T. Petrie. Une variante de cette approche, due elle aussi à Jelinek et ses collègues a trouvé un regain d'intérêt dans le domaine des codes correcteurs d'erreurs en transmissions numériques où elle est utilisée dans la technique dite des turbocodes.

La propriété importante des processus markoviens est que l'évotution de l'automate après l'instant t ne dépend que de la valeur de l'état où il se trouve à cet instant et des commandes qui lui sont appliquées ensuite et non de ce qu'il a subi avant d'arriver à cet état. En particulier, le futur ne dépend pas de façon dont l'automate s'est retrouvé dans l'état en question.

Les M états et les N valeurs possibles des mesures ainsi que les probabilités $a(m,m')$, $b(m,n)$ et $d(m)$ caractérisent le modèle.

Nous sommes amenés à traiter trois problèmes

1 - Reconnaissance

On a observé $Y = [y_0, \ldots, y_t, \ldots, y_T]$. $[a(m,m'), b(m,n), d(m)]$ est donné. Quelle est la séquence d'état $S = [s_0, \ldots, s_T]$ la plus probable qui a engendré $[y_0, \ldots, y_t, \ldots, y_T]$?

2 - Probabilité d'observation d'une séquence

On a observé une séquence de mesures $Y = [y_0, \ldots, y_t, \ldots, y_T]$. Quelle est la probabilité pour que l'automate caractérisé par les paramètres $[a(m,m'), b(m,n), d(m)]$ ait engendré cette séquence?

3 - Apprentissage

On a observé $[y_0, \ldots, y_t, \ldots, y_T]$. Comment calculer (ou plutôt actualiser les paramètres du modèle $[a(m,m'), b(m,n), d(m)]$ pour maximiser la probabilité d'observer $[y_0, \ldots, y_t, \ldots, y_T]$?

Nous commencerons par traiter le problème de la reconnaissance pour donner ensuite les formules donnant la probabilité d'observation d'une séquence. Puis ces formules seront utilisées pour répondre à la troisième question, l'apprentissage.

2.2 Algorithme de Viterbi pour la reconnaissance d'une séquence

Cet algorithme a pour but de trouver la séquence d'états la plus probable ayant produit la séquence mesurée $[y_0, \ldots, y_T]$.

A l'instant (t) on calcule par récurrence pour chacun des états

$$r_t(m) = \max p(s_0, \ldots, s_{t-1}, \mathbf{s_t} = \mathbf{m}, y_0, \ldots, y_t). \tag{9}$$

Le maximum étant calculé sur toutes les séquences d'états possibles $[s_0, \ldots, s_{t-1}]$

- *Initialisation*:

  A l'instant $t = 0$

$$r_0(m) = d(m)b(m, y_0). \tag{10}$$

- *Récurrence*:

On suppose qu'à l'instant $(t-1)$ on a calculé $r_{t-1}(m)$ pour chacun des M états. On a alors

$$r_t(m') = \max_m r_{t-1}(m)a(m,m')b(m',y_t). \tag{11}$$

L'état m le plus probable occupé à l'instant $t-1$ à partir duquel l'automate a évolué vers l'état m' à l'instant t est l'état tel que $r_{t-1}(m)a(m,m')b(m',y_t)$ est maximum.

Pour chacun des états m', on calcule ainsi $r_t(m')$; chacun de ces états a un prédécesseur $q_t(m')$. Ce prédécesseur pourra servir à retrouver la séquence d'état la plus probable ayant engendré les mesures $[y_0,\ldots,y_T]$.

• *Fin de l'algorithme*:

L'état f_T retenu à l'instant T est celui pour lequel $r_T(m)$ est maximum. La probabilité pour la séquence mesurée ait été émise par l'automate est $r_T(m)$.

On peut retrouver la séquence des états en retrouvant le prédécesseur de f_T

$$f_{T-1} = q_{T-1}(f_T), \tag{12}$$

et récursivement

$$\begin{aligned} f_{T-2} &= q_{T-2}(f_{T-1}),\\ f_{T-3} &= q_{T-3}(f_{T-2}),\\ &\cdots\\ f_1 &= q_1(f_2),\\ f_0 &= q_0(f_1). \end{aligned} \tag{13}$$

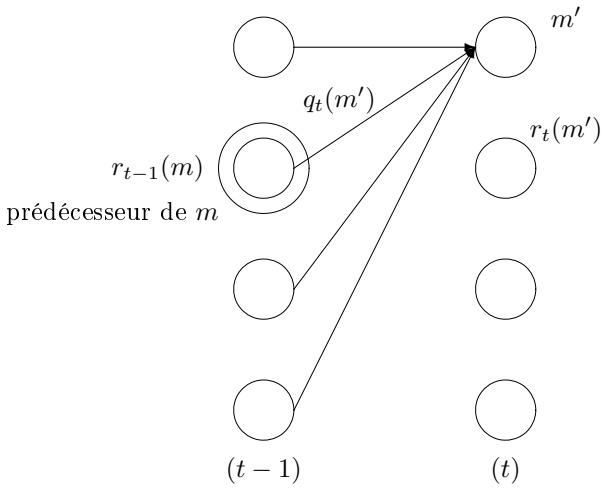

Figure 2: Sélection d'un chemin dans le treillis entre les instants $t-1$ et t.

On peut ainsi calculer pour chacun des modèles de Markov représentant par exemple un mot du vocabulaire à reconnaitre la probabilité que la séquence mesurée étudiée ait été engendrée par cet automate puis comparer les résultats.

Une présentation plus détaillée de l'algorithme de Viterbi utilisé pour la correction d'erreurs en transmissions numériques se trouve dans les notes de cours sur la transmission (cf. www.essi.fr/ leroux/).

2.3 Probabilité d'observation d'une séquence

La probabilité qu'une séquence d'états $S=[s_0,\ldots,s_t,\ldots,s_T]$ ait engendré Ys'obtient en utilisant la propriété des sources markoviennes

$$p(Y/S) = p([y_0,\ldots,y_T]/[s_0,\ldots,s_T]), \tag{14}$$

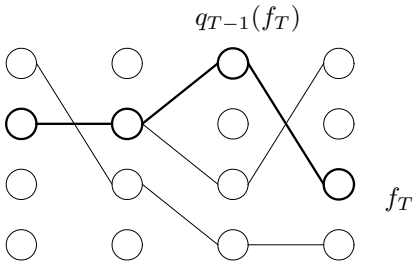

Figure 3: Reconstuction du chemin correspondant à la séquence optimale $[s_0, \ldots, s_T]$ si elle a été engendrée par l'automate.

$$p(Y/S) = \prod_{t=0}^{T} p(y_t/s_t), \tag{15}$$

$$p(Y/S) = \prod_{t=0}^{T} b(s_t, y_t). \tag{16}$$

Par ailleurs

$$p(S) = p(s_0, \ldots, s_T) = d(s_0) \prod_{t=1}^{T-1} a(s_{t-1}, s_t). \tag{17}$$

Par conséquent

$$p(Y, S) = d(s_0)b(s_0, y_0) \prod_{t=1}^{T-1} a(s_{t-1}, s_t)b(s_t, y_t), \tag{18}$$

et la probabilité d'avoir émis la séquence Y est donnée par la sommation sur toutes les M^T séquences S possibles

$$p(Y) = \sum_{S} d(s_0)b(s_0, y_0) \prod_{t=1}^{T-1} a(s_{t-1}, s_t)b(s_t, y_t), \tag{19}$$

formule inutilisable en pratique car elle nécessite de l'ordre de $T \times M^T$ opérations. On peut réduire le nombre d'opérations en effectuant un calcul par récurrence.

- **Calcul à t croissant**
  On définit

$$\alpha_t(m) = p(s_t = m, [y_0, \ldots, y_t]), \tag{20}$$

$$\alpha_0(m) = d(m,)b(m, y_0). \tag{21}$$

  Compte tenu des propriétés des processus markoviens, on peut écrire

$$\alpha_{t+1}(m) = \sum_{m'=1}^{M} \alpha_t(m')a(m', m)b(m, y_{t+1}). \tag{22}$$

On peut alors écrire $p(y_0, \ldots, y_T)$ en fonction de $\alpha_T(m)$

$$p(Y) = \sum_{m} \alpha_T(m), \tag{23}$$

ce qui ne nécessite que $2M^2T$ opérations.

- **Calcul à t décroissant**
  De même, on peut calculer une récurrence dans le sens rétrograde (temps décroissant) sur les

$$\beta_m = p(y_{t+1}, \ldots, y_T/s_t = m), \tag{24}$$

avec comme condition initiale

$$\beta_T(m) = 1. \tag{25}$$

Cette récurrence s'écrit

$$\beta_t(m) = \sum_{m''} a(m, m'')b(m'', y_{t+1})\beta_{t+1}(m''). \tag{26}$$

• **Probabilité de passage dans un état**
Soit

$$\lambda_t(m) = p(s_t = m, [y_0, \ldots, y_T]), \tag{27}$$

$$\lambda_t(m) = p(s_t = m/[y_0, \ldots, y_T]) \times p([y_0, \ldots, y_T]). \tag{28}$$

On peut montrer que

$$\lambda_t(m) = \frac{\alpha_t(m)\beta_t(m)}{\sum_{m'} \alpha_t(m')\beta_t(m')}. \tag{29}$$

L'état le plus probable à l'instant t est l'état $\hat{s}_t = m$ tel que $\lambda_t(m)$ est maximum.

Remarques
Les démonstrations des propriétés sont données dans le document de cours sur les turbocodes (cf. www.essi.fr/ leroux/). Elles ne sont pas reprises dans ce document. Les différences portent sur deux points: lorsqu'on traite les codes correcteurs d'erreurs, il n'est pas nécessaire de normaliser les calculs itératifs car on considère qu'on connait la probabilité d'observation d'une séquence; la seconde différence est due au fait que dans le cas des codes correcteurs d'erreurs il semble plus pratique d'associer les émissions de données aux transitions entre états et non aux états; en conséquences les facteurs de la forme $a(m', m)b(m, n)$ apparaissant dans les formules sont remplacés par des facteurs du type $\gamma_t(m', m)$:

$$\gamma_t(m', m) = p((s_t = m, y_t)/s_{t-1} = m'). \tag{30}$$

2.4 Réestimation des paramètres d'un modèle pour l'apprentissage

On définit

$$\sigma_t(m', m) = p(s_{t-1} = m', s_t = m, [y_0, \ldots, y_T]), \tag{31}$$

et on peut montrer que

$$\sigma_t(m', m) = \frac{\alpha_{t-1}(m')a(m', m)b(m, y_t)\beta_t(m)}{\sum_{m_1=1}^{M}\sum_{m_2=1}^{M} \alpha_{t-1}(m_1)a(m_1, m_2)b(m_2, y_t)\beta_t(m_2)}. \tag{32}$$

La probabilité

$$\lambda_t(m) = p(s_t = m, [y_0, \ldots, y_T]), \tag{33}$$

peut s'écrire

$$\lambda_t(m) = \sum_{m'=1}^{M} \sigma_t(m', m). \tag{34}$$

Réestimation des paramètres $[a(m', m, b(m, n), d(m)]$
Nombre moyen de débuts de la séquence dans l'état m:

$$\hat{d}(m) = \lambda_0(m). \tag{35}$$

$\hat{a}(m, m')$ est le rapport entre le nombre moyen de transitions $m \to m'$ et le nombre de passage dans l'état m

$$\hat{a}(m, m') = \frac{\sum_{t=0}^{T-1} \sigma_t(m, m')}{\sum_{t=0}^{T-1} \lambda_t(m)}. \tag{36}$$

$\hat{b}(m,n)$ est le rapport entre le nombre moyen de fois où on observe n dans l'état m et du nombre de fois où l'automate s'est trouvé dans l'état m.

$$\hat{b}(m,n) = \frac{\sum_{t=0}^{T-1} \lambda_t(m)\delta(y_t - n)}{\sum_{t=0}^{T-1} \lambda_t(m)}. \tag{37}$$

Dans la sommation du numérateur, $\delta(y_t - n)$ vaut un lorsque $y_t = n$ et zéro dans le cas contraire.

3 Mise en œuvre

La mise en œuvre nécessite des étapes préalables, en particulier la représentation sous forme de mesures quantifiées des données extraites du signal de parole.

3.1 Analyse spectrale à court terme

On dispose au départ d'une analyse temps fréquence du signal vocal, (l'évolution du spectre du signal au cours du temps); chacun de ces spectres peut être représenté par un vecteur d'une dizaine de paramètres (voir le chapitre consacré à la prédiction linéaire dans le cours de traitement numérique du signal (cf. www.essi.fr/ leroux/)).

La plupart des documents consacrés à l'application des HMM à la reconnaissance de la parole utilisent les données "cepstrales" pour représenter ce spectre à cours terme. Le "cepstre" est la transformée de Fourier inverse du logarithme de la densité spectrale du signal : on n'en conserve que les premiers échantillons qui correspondent aux variations lentes de la densité spectrale, c'est à dire la forme générale du spectre, et qui contient en particulier les informations sur les formants ; tandis que les échantillons d'ordre plus élevé correspondent aux variations rapides de la densité spectrale en fonction de la fréquence. Ces derniers échantillons sont en général caractéristiques des harmoniques de la fréquence fondamentale des cordes vocales et donc de l'intonation. Ils ne sont pas pris en compte dans le processus de reconnaissance.

Pour calculer les coefficients cepstraux, Davis et Mermelstein ont préconisé l'emploi d'un banc de 20 filtres: ces filtres sont des filtres de réponse en fréquence triangulaire dont la largeur de bande est telle que la somme des réponses en fréquence est constante; la fréquence centrale suit une échelle "MEL": 10 filtres échelonnés linéairement entre 100 et 1000 Hz, échelonnage exponentiel des 10 autres filtres entre 1000 Hz et 4000 Hz (1149, 1320, 1516, 1741, 2000, 2297, 2639, 3031, 3482, 4000.) On mesure l'énergie en sortie de chacun de ces 20 filtres, soit $C(n)$ pour $n = 1, \ldots, 20$ (il est préférable de faire ce calcul dans le domaine des fréquences plutôt que de programmer les filtres dans le domaine temporel); puis on en calcule le logarithme et la transformée de Fourier inverse de cette séquence

$$c(k) = \sum_{n=1}^{20} \log\left(C(n)\right) \cos\left[\frac{\pi}{20} k \left(n - \frac{1}{2}\right)\right], \tag{38}$$

pour $k = 1, \ldots, K$. Le nombre de coefficients cepstraux (K) peut être réduit à six.

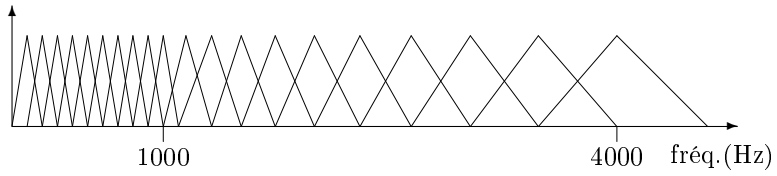

Figure 4: Banc de 20 filtres proposé par Davis et Mermelstein.

Il est possible d'établir un lien entre les paramètres du filtre de synthèse récursif calculé par la méthode de la prédiction linéaire $[a_0, \ldots, a_p]$

$$A(z) = \frac{1}{a_0 + a_1 z^{-1} + a_p z^{-p}}, \tag{39}$$

et les coefficients cepstraux $[c_0, \ldots, c_k]$:

$$c(k) = -a(k) - \sum_{\ell=1}^{k-1}(1 - \frac{\ell}{k})a(\ell)c(k-\ell). \tag{40}$$

(Voir par exemple l'ouvrage de R. Boite et M. Kunt, "Traitement de la parole"). Toutefois les expériences menées par Davis et Mermelstein montrent que le calcul fondé sur les bancs de filtres et la transformée de Fourier donnent de meilleurs résultats, en particulier dans l'analyse des transitoires rapides comme le passage d'une plosive à une voyelle.

Il faut ensuite quantifier ces vecteurs permettant de décrire le spectre à court terme de la parole pour leur faire correspondre un point dans l'espace des mesures : le nombre de mesures possibles est fini. On peut toutefois étendre l'approche à des cas moins contraignants où on recalcule pour chaque état la probabilité $b(m, y_t)$ d'obtenir la mesure y_t. Pour cela on représente les mesures par des densités de probabilités gaussiennes dont on estimera la moyenne et la covariance lors de l'apprentissage.

3.2 Quantification vectorielle

Les vecteurs de données issus des mesures ont des composantes réelles. Pour les traiter sous la forme d'un modèle de Markov caché, il faut les quantifier. La technique utilisée, la quantification vectorielle a été proposée par R. M. Gray et ses collègues.

L'apprentissage se fait de manière itérative : on se donne N vecteurs qui sont répartis parmi le grand nombre de vecteurs $v_1, \ldots, v_L$ qu'il faut quantifier. Il est utile d'utiliser l'information dont on dispose pour choisir ces vecteurs ; en l'absence d'information, on peut se contenter de les choisir au hasard, mais cela peut avoir des conséquences fâcheuses sur la qualité du résultat obtenu. Ces N vecteurs sont associés à des points $g_1^0, \ldots, g_N^0$ dans l'espace $\mathbf{R}^N$.

On assigne à chacun des "centres de classe" g_k^0 les vecteurs v_ℓ les plus proches de g_k^0 :
v_ℓ appartiendra à la classe G_k caractérisée par g_k^0 si pour tout $k' \neq k$

$$||v_\ell - g_k^0|| < ||v_\ell - g_{k'}^0|| \tag{41}$$

où $||v||$ est une norme, par exemple la norme euclidienne. On calcule ensuite le centre de gravité des vecteurs v_ℓ appartenant à G_k, soit g_k^1 et on réitère les deux opérations : assignation des données à une classe puis réévaluation du centre de gravité des classes jusqu'à convergene. La solution obtenue par cette méthode (ou une de ses nombreuses variantes) n'est peut-être pas optimale mais les défauts qu'elle peut engendrer sont paliés par la représentation probabiliste des mesures dans le modèle de markov caché qui leur sera ensuite appliqué.

Dans certaines implémentations, on n'explicite pas les erreurs de mesure comme une probabilité $b(m, n)$ associées à un modèle de Markov caché, mais comme une probabilité d'erreur de mesure (supposée gaussienne) entre la mesure y_t et la donnée qui serait théoriquement émise lorsque l'automate est dans l'état associé à la classe G_k. L'apprentissage des $b(m, n)$ est alors remplacé par l'estimation des paramètres des densités de probabilités gaussiennes multivariables considérées.

3.3 Forme usuelle de l'automate

En général un automate est associé à une séquence type et les défauts qui affectent le processus de reconnaissance sont essentiellement des différences de rythmes. Ces différences peuvent se traduire par une répétition du même état (la séquence analysée est énoncée plus lentement que la séquence retenue par apprentissage) ou bien par un saut d'un état à l'autre en omettant l'état intermédiaire de la séquence apprise. C'est le schéma du modèle de Bakis.

4 Bibliographie

[1] R. Bahl and F. Jelinek, "Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition, ", IEEE trans. Inform. Theory, vol. IT-21, pp 404-411, 1975.

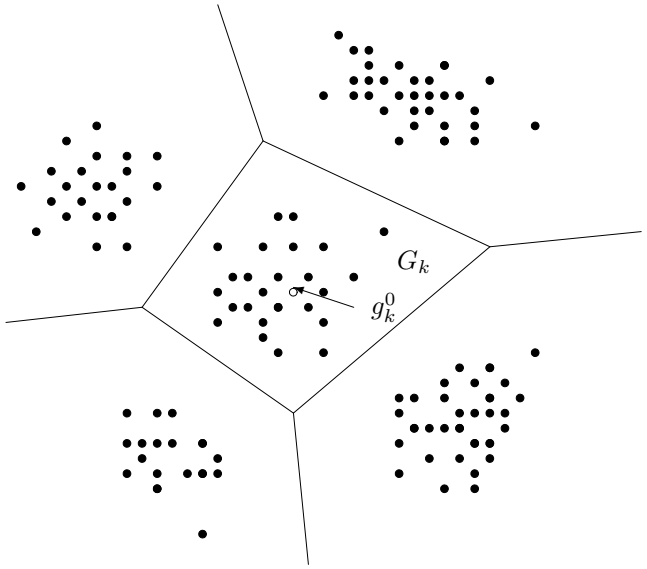

Figure 5: Définition des classes de données en quantification vectorielle.

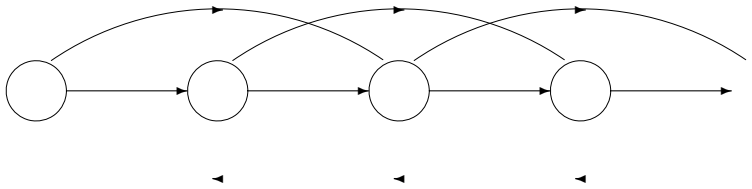

Figure 6: Automate de Bakis souvent utilisé pour représenter l'évolution des états d'un modèle de Markov caché en reconnaissance de parole.

[2] L. E. Baum and T. Petrie, Statistical inference for probabilistic functions of ninite state Markov chains, Ann. Math. Stats., vol. 37, pp. 1554-1563, 1966.

[3] R. Boite et M. Kunt, Traitement de la parole, Presses polytechniques romandes, 1987.

[] Calliope, La parole et son traitement automatique - - Collection CNET - ENST - Masson

[] S. B. Davis and P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, IEEE trans. on ASSP, vol ASSP 28, no 4, August 1980, pp 357-366.

[4] F. Jelinek, Statistical Methods for Speech Recognition, MIT PRESS, 1998.

[5] Y. Linde, A. Buzo and R. M. Gray, An algorithm for vector quantizer design, IEEE trans. comm. COM-28, no 1, January 1980, pp.84-95.

[6] L. R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition, Proceedings of the IEEEE, vol. 77, no 2, Feb 1989.

[7] L. R. Rabiner et Juang Fundamentals of speech recognition, Prentice Hall PTR 1993.

[8] The Hidden Markov Model Toolkit (HTK) : `http://htk.eng.cam.ac.uk/`