Dynamic Bayesian Networks for Speaker Verification

Eduardo Sanchez-Soto

École Doctorale
d'Informatique,
Télécommunications
et Électronique de Paris

Thèse

présentée pour obtenir le grade de docteur
de l'École Nationale Supérieure des Télécommunications

Spécialité : **Signal et Images**

Eduardo SÁNCHEZ SOTO

Réseaux Bayésiens Dynamiques pour la Vérification du Locuteur

Soutenue le 10 mai 2005 devant le jury composé de

Jean Paul HATON	Président
Régine ANDRÉ-OBRECHT	Rapporteurs
Jean-François BONASTRE	
Chafic MOKBEL	Examinateurs
Paul MUNTEANU	
Guillaume GRAVIER	
Gérard CHOLLET	Directeur de thèse
Marc SIGELLE	

REMERCIEMENTS

Je remercie avant tout le peuple du Mexique qui, sans le savoir, m'a prêté de l'argent pour financer cette thèse pendant trois année grâce à une bourse du CONACYT (Conseil National de Science et Téchnologie du Mexique).

J'aimerais ensuite remercier tous les membres du jury. Particulièrement, je tiens à remercier Madame Régine André-Obrecht et Monsieur Jean François Bonastre d'avoir accepté le travail de rapporteur cette mémoire, Monsieur Chafic Mokbel pour son aide et ses conseils scientifiques précieuse, Monsieur Jean Paul Haton pour l'intérêt porté sur cette travail et Monsieur Guillaume Gravier et Monsieur Paul Munteanu pour les remarques et commentaires.

Cette thèse n'aurait pu voir le jour sans l'aide de mes directeurs de thèse : Monsieur Gérard Chollet et Monsier Marc Sigelle. Je leur remercie de m'avoir permis réaliser cette thèse et de la liberté que m'ont accordé pour gérer le travail tout au long des années. Particulièrement à Monsieur Gérard Chollet pour les conseils scientifiques et à Monsieur Marc Sigelle pour la rigueur mathématique.

Egalement, je voudrias remercier à Maurice Charbit pour les réflexions, et l'aide.

J'ai eu egalement l'aide précieuse des plusieurs amis. Je remercie très sincèrement à Raphaël, Slim, Thomas, Chloé, Julie et Khalid sans qui cette memoire ne aurait pas été la même. Je garderai un souvenir précieux de ces années et des collègues et amis qui m'ont accompagnés pendant cette période.

Un grand merci à Catherine Vazza, Patricia Friedrich et Laurence Zelmar pour le soutien.

Un grand merci aussi à Jean François Bonastre pour toute l'aide.

Finalement, merci Lola ! Sache que tu as été mon soutien le plus fort !

ABSTRACT

Dynamic Bayesian Networks
for Speaker Verification
by
Eduardo Sánchez Soto
Doctor, Speciality in Signal and Image
École Nationale Supérieure des Télécomunications, PARIS
Professor Gérard CHOLLET, Chair
Associate Professor Marc SIGELLE

This thesis is concerned with the statistical modeling of speech signal applied to Speaker Verification using Bayesian Networks. The main idea of this work is to use Bayesian Networks as a mathematical tool to model pertinent speech features keeping its relations. It combines theoretical and experimental work on Speaker Verification.

The performance of state-of-the-art speech processing systems is still far from that of humans. The difference between systems and humans is the quantity of information and the relationships between the sources of information used to make decisions. From speech signal, the mood, emotive state and identity present in spectral and prosodic features can be combined to improve performances. A single statistical framework that keeps the conditional dependence and independence relations between those variables is difficult to attain. To some degree this is caused by the lack of good statistical models. Therefore, in this work the use of Bayesian Networks as a tool for modeling the available information and their independence and dependence relationships is proposed for Speaker Verification.

The first part of this thesis reviews the main modules of a SV systems, the possible sources of information in the speech signal as well as the basic concepts of graphical models, specially their representation. Directed Acyclic Graphs receive particular attention because they are the main probabilistic tools used in this work. Hidden Markov Models and their variants are studied from the point of view of Dynamic Bayesian Networks.

The second part of this thesis deals with Modeling in the proposed SV systems. A new approach to the problems associated with SV is proposed. The problem of inference and learning (parameters and structure) in BN are presented. It is described how to learn the relations of conditional independence among the variables directly from the data in order to obtain a structure. The Speaker Verification system developed here uses learning techniques to retrieve the conditional independencies between the sources of information. These relations are then used in order to build an adapted Bayesian Network. Even if the variables space is fully observable, the structure space, the search algorithm and the measurement score were considered. The technique used works in a limited structure space. In particular, a new model adaptation technique for Speaker Identification has been proposed. This adaptation is based on a measure between Conditional Probability Distributions for discrete variables, as well as, on regression matrix for continuous variables used to model the relationships conditional dependencies of this approach based on Bayesian Networks. In a large data base for the Speaker Verification task, the results have confirmed the potential of use Bayesian Networks approach.

Contents

List of Figures

List of Tables

Chapter 1

Résume Étendu

1.1 Présentation du Problème

Le but des systèmes de Vérification du Locuteur (VL) est de confirmer l'identité proclamée en utilisant de l'information disponible provenant du locuteur. Fondamentalement, la parole est un des moyens les plus employés par les humains pour communiquer. Par conséquent, c'est une source primordiale de renseignements. De plus, dans certaines applications, la parole est la seule source d'information disponible ; comme c'est le cas par exemple dans un appel téléphonique. Quelques secondes de parole constituent une quantité très importante d'information. La plus importante parmi ces informations est le message. Mais, en plus de ce message, une autre information très caractéristique est également présente dans le signal acoustique : l'identité du Locuteur.

La performance des systèmes réels de VL est encore loin de celle des humains. Un problème majeur dans les systèmes de VL est de trouver l'ensemble de caractéristiques qui représentent le mieux chacun des Locuteurs. Une différence fondamentale entre les systèmes de VL et les humains est la quantité et la qualité de l'information utilisée ainsi que la relation entre les sources d'information employées pour prendre des décisions. Le message et l'identité du Locuteur sont codés dans plusieurs niveaux d'abstraction. Chaque personne a une voix différente, une manière et un rythme différents de parler, une tonalité de voix différente, quelques mots préférés, etc. Du niveau acoustique au niveau linguistique et paralinguistique, la parole aide à communiquer certaines intentions qui sont codées et exprimées dans une phrase. Pour une même phrase, le signal de parole associé à un locuteur particulier est unique. Au niveau acoustique, par exemple, le spectre pourrait aider quelqu'un ou un système de VL à identifier son interlocuteur. Mais pour cette tâche, les humains n'emploient pas que le spectre ou l'information acoustique dans une communication normale et emploient habituellement toute l'information disponible : la tonalité de la voix, les mots utilisées, etc. Ils emploient également des données prosodiques, aussi bien que des caractéristiques segmentales et suprasegmentales comme l'intonation, l'accent, la fréquence fondamentale (et la façon de parler). Cependant, n'importe laquelle de ces informations n'est pas en soi suffisante pour faire une distinction entre deux personnes différentes. Toutes ces données doivent donc être utilisées pour caractériser l'identité d'un Locuteur.

En plus de ces différences, quelques problèmes dus à l'exploitation du système dans des conditions réelles peuvent être identifiés. Les conditions de prise de son, au niveau du matériel et de l'environnement acoustique, doivent être prises en considération afin de concevoir un système robuste. L'environnement est un problème difficile à gérer pour le bon fonctionnement des systèmes de VL. En plus de l'information provenant de la parole, le signal contient alors également le bruit de l'environnement. En outre, la parole est modifiée par le canal de communication avant d'arriver au récepteur final. Le signal acoustique contient, en plus des informations propres au Locuteur, des caractéristiques de la voie de transmission. Toutes ces dégradations du signal rendent le problème plus complexe.

En pratique, la différence entre les environnements d'apprentissage et de test pose un autre problème. Cette dissimilitude est un problème proche du problème du manque de données. En effet, ces deux problèmes sont reliés entre eux car le fait que tous les environnements possibles ne sont pas représentés dans la phase d'apprentissage équivaut à un manque de données. Les échantillons de parole obtenus à partir de chaque Locuteur ne vont jamais représenter toutes les conditions potentielles d'utilisation. Il est peu réaliste de pouvoir obtenir des échantillons provenant de tous les environnements possibles, de tous les états d'acquisition des signaux et de tous les canaux de transmission.

Comme obstacle supplémentaire pour les systèmes de VL, nous pouvons mentionner le fait bien connu du changement de la voix des humains avec le temps. En général, l'application de système de VL dans de vraies conditions doit prendre en compte toutes les différences possibles entre les échantillons utilisés pour l'apprentissage et les données utilisées pour le test, étant donné que ces différences sont source de complications et de problèmes.

Pour surmonter toutes ces difficultés, les développements actuels dans les systèmes de VL emploient plusieurs techniques qui compensent ces différences afin d'obtenir de meilleurs résultats à partir de systèmes plus robustes. Fondamentalement, on distingue trois techniques. La première, appelée technique de normalisation, essaie de rendre l'information utilisée indépendante des conditions d'utilisation. La deuxième, la techniques d'adaptation, essaie d'adapter la connaissance acquise de chaque Locuteur aux nouvelles conditions d'environnement et d'utilisation. Et finalement, la dernière technique consiste à employer une connaissance *a priori* pour équilibrer le manque de données dans la nouvelle information.

Dans ce travail, nous nous adressons à certains de ces problèmes en utilisant deux approches différentes qui reflètent nos contributions. D'abord, pour améliorer les performances des systèmes de VL nous proposons d'employer un système basé sur des Réseaux Bayesiens (RBs) afin de combiner plusieurs sources d'information. Ce système permet d'intégrer l'information provenant de différentes sources dans un cadre statistique simple qui garde les relations conditionnelles de dépendance et d'indépendance entre toutes ces données. En second lieu, pour rendre le système plus robuste aux différentes conditions de l'utilisation, nous suggérons d'employer des techniques d'adaptation utilisant une connaissance *a priori*. Enfin, nous proposons une technique pour adapter les RBs basée sur certaines caractéristiques mathématiques des relations d'indépendance conditionnelles de ce type de modèles.

1.2 Systèmes de Vérification du Locuteur

Les systèmes de SV se divisent en deux phases différentes. La première est la phase d'apprentissage et la seconde est celle de test. Dans la phase d'apprentissage, le modèle d'un Locuteur est produit en se servant de quelques phrases prononcées par celui-ci. Dans la phase de test, les modèles produits dans les premières phases sont employés pour vérifier l'identité proclamée pour un échantillon de parole. Un système automatique représentatif de Vérification de Locuteur (VAL) se compose de quatre modules principaux comme cela est représenté sur la Figure F.1 Le premier module est chargé de l'obtention et de la numérisation du signal acoustique. Cette tâche est réalisée par filtrage et en employant un convertisseur analogique numérique.

Le deuxième module, celui d'extraction de paramètres, est consacré à l'obtention d'informations convenables pour la VL. La parole est le résultat de différentes transformations qui ont lieu à différents niveaux (sémantique, linguistique, articulatoire et acoustique). A chacun de ces niveaux, nous pouvons obtenir des informations à propos d'un Locuteur donné. Habituellement, pour extraire ces informations, le signal acoustique est d'abord divisé en intervalles (généralement entre 10 et 30 ms) appelés trames. Dans la plupart des systèmes réels, l'information spectrale est employée ainsi que sa dynamique qui est représentée par la première et la deuxième dérivée (appelées coefficients Δ et $\Delta\Delta$). En plus des caractéristiques spectrales de la parole, d'autres informations peuvent être obtenues comme la fréquence fondamentale, qui est un exemple d'information prosodique.

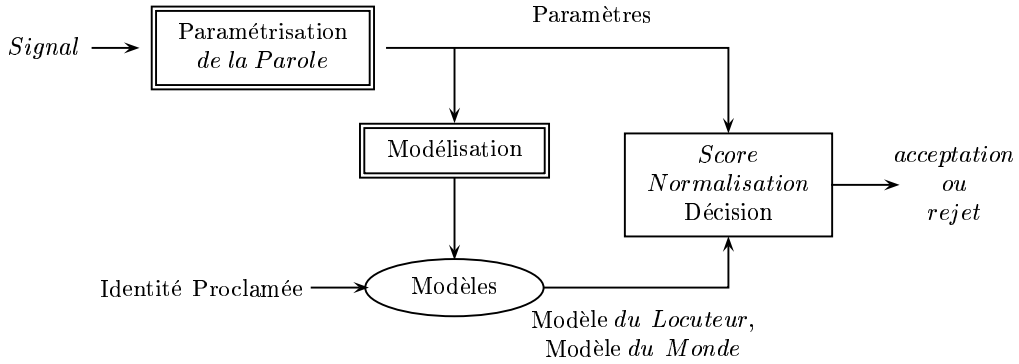


Figure F. 1.1: *Modules principaux d'un système de vérification automatique du Locuteur. Notre travail a porté sur les modules représentés dans des doubles carrés.*

Le troisième module, modélisation, est un des plus importants dans les systèmes de VL. L'approche statistique est de nos jours la plus utilisée. Ces modèles sont basés principalement sur les Modèles de Markov Cachés (MMC), les Modèles de Mélange de Gaussiennes (MMG). Dans ce travail, nous proposons l'utilisation de réseaux bayesiens (RB) qui estiment la fonction de densité de probabilité d'un ensemble de variables.

Afin d'obtenir de bons modèles, il faut employer la quantité nécessaire de signaux de parole reflétant les circonstances prévues d'utilisation. Si cette contrainte n'est pas satisfaite, des techniques d'adaptation peuvent être utilisées pour surmonter le problème. Les techniques d'adaptation proposées pour les RB sont une des contributions de ce travail et elles seront abordées ultérieurement. Quand il n'y a pas assez d'échantillons de parole d'un locuteur donné, le modèle peut être obtenu en adaptant un bon modèle générique. Cette adaptation est faite en utilisant des techniques bayesiennes comme celle du maximum a posteriori (MAP) [Mokbel, 2001; Gauvain and Lee, 1994] ou celle de la régression linéaire du maximum de vraisemblance (MLLR).

Le dernier module de la chaîne d'un système de VL est la décision. Dans les modèles statistiques, les scores sont basés sur le calcul de vraisemblance. Le problème de décision dans un système de VL peut être vu comme un problème de classification à deux classes. La première classe correspond au locuteur S et la deuxième classe correspond à un état appelé le non-locuteur $\overline{S}$. En utilisant la théorie de classification bayesienne on peut justifier l'utilisation du rapport de vraisemblance, ou le logarithme de ce rapport pour prendre la décision finale :

$$S(x) = \log \frac{P(x|S)}{P(x|\overline{S})} \; > \; \log(seuil) = constant.$$

La décision est prise en comparant le rapport de vraisemblance à un seuil. Cette règle est basée sur les vraies fonctions de densité de probabilité. Or, dans une application réelle ces vraies fonctions ne sont pas connues. Les fonctions calculées dépendent des conditions d'utilisation réelles, de la phrase prononcée et également de chaque Locuteur. Par conséquent, les valeurs obtenues doivent être normalisées. La première technique de normalisation est conçue pour compenser la longueur de la phrase. Pour le score $S(x)$ d'une phrase x de longueur T, le score normalisé $\widehat{S}(x)$ est :

$$\widehat{S}(x) = \frac{1}{T} \; \log \frac{P(x|S)}{P(x|\overline{S})}.$$

La variabilité des scores peut venir d'autres sources que de la longueur de la phrase comme par exemple du canal de transmission. D'autres techniques de normalisation sont alors employées pour ajuster le score étant donné ces autres sources de fluctuation.

Performance des systèmes de VL

Étant donné qu'un système de SV doit vérifier l'identité proclamée d'un locuteur donné, deux types d'erreurs peuvent être commis. La première erreur est d'accepter un imposteur, appelée fausse acceptation (*FA*). La deuxième erreur est de rejeter un client, appelée faux rejet (*FR*). Ces erreurs sont exprimées par les taux de FA et de FR qui sont évalués respectivement par :

$$\begin{aligned} FAR &= \frac{\#FA}{\#\text{accès imposteur}}, \\ FRR &= \frac{\#FR}{\#\text{accès client}}. \end{aligned}$$

Les performances des systèmes de VL sont représentées par des courbes de Différence de Détection d'Error (DET). Les deux erreurs possibles d'un système de VL sont tracées dans la courbe en fonction du seuil de décision utilisé. Si les scores des clients et des imposteurs suivent une distribution normale la courbe doit être une ligne.

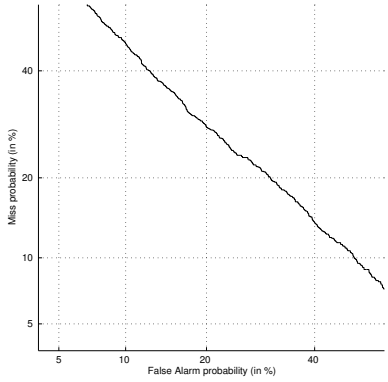


Figure F. 1.2: *Exemple d'une courbe DET.*

1.3 Sources d'Information pour La Vérification du Locuteur

Pour détailler la chaîne de traitement d'un système de VL, nous commençons par décrire la paramétrisation de la parole et l'obtention des sources d'information à partir du signal acoustique. Dans un signal de parole, on peut trouver beaucoup d'information en plus du message, comme des informations sur le Locuteur et son identité. Dans la plupart des systèmes de VL, seule l'information spectrale est prise en compte. Généralement, la prosodie et les caractéristiques suprasegmental, comme l'intonation, l'accent ou la fréquence fondamentale ne sont pas pris en compte. Un système plus robuste devraient employer toutes ces données qui caractérisent un Locuteur. La difficulté qui se présente ici est de savoir comment combiner efficacement ces données. Dans nos travaux, il est proposé de le faire au moyen de RB.

Résiduel de l'analyse en Prédiction Linéaire

Chez les humains, la parole est produite par des organes dont l'anatomie et le contrôle moteur est spécifique à chacun des Locuteurs. L'air traverse la glotte, les cordes vocales et le larynx. Par conséquent le signal

d'excitation qui en résulte caractérise le Locuteur et peut être une source supplémentaire d'information. L'analyse en Prédiction linéaire (PL) fournit une méthode pour séparer l'information du conduit vocal et celle issue de l'excitation. Le résiduel de la PL [Thévenaz, 1993; Faúndez-Zanuy and Rodríguez-Porcheron, 1998] est particulièrement plat, et l'effet de filtrage du conduit vocal peut être enlevé en créant un filtre inverse. Ce filtre est obtenu en utilisant les coefficients de PL calculés avec la fonction d'autocorrélation et l'algorithme de Yule- Walker :

$$\text{résiduel}(n) = G\Big(s(n) - \hat{s}(n)\Big) \ \forall n,$$

où, G est un facteur de gain, $s(n)$ les échantillons de parole et $\hat{s}(n)$ les échantillons obtenus avec l'analyse PL. La Figure ci-dessous montre un signal voisé typique ayant une structure périodique caractéristique en haut et au dessus le signal résiduel correspondant. On peut voir que la structure périodique de la parole est représentée comme un train d'impulsions avec une fréquence constante dans le résiduel de PL.

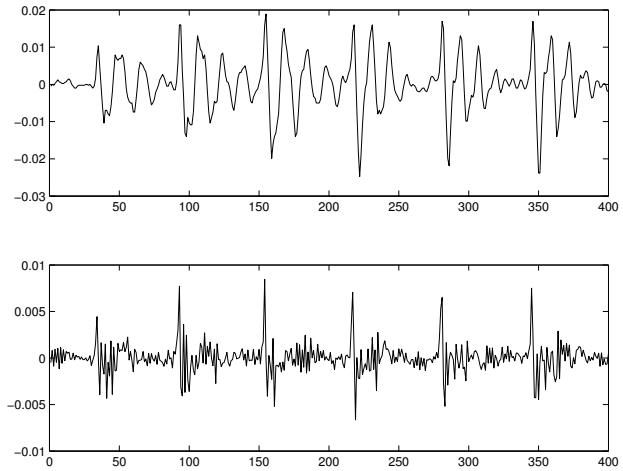


Figure F. 1.3: *Exemple d'un signal voisé en haut et le résiduel correspondant en dessous.*

Information prosodique

La prosodie est une autre caractéristique très importante de la parole. Quand on parle d'information prosodique, on désigne en fait des caractéristiques de la parole dont le support n'est pas qu'un segment phonétique simple, mais une plus grande unité, comme un mot ou une phrase entière. La prosodie est une caractéristique qui renseigne sur la structure linguistique de la parole. La fréquence fondamentale est une des informations prosodiques les plus utilisées.

La fréquence fondamentale, ou pitch, d'un son périodique est une composante sinusoïdale qui a la même période que le signal original. La fréquence fondamentale n'est pas la hauteur tonale perçue et subjective d'un son complexe. La fréquence fondamentale est une caractéristique du signal qui a une relation directe avec les mouvements de la glotte. Parmi plusieurs méthodes, celle de l'autocorrélation est l'une des plus simples et des plus rapides à effectuer pour mesurer le pitch. Un exemple d'un signal de parole et du pitch correspondant est donné dans la Figure ci-dessous.

Information Spectral

En général, la source d'information la plus utilisée et la plus fiable en traitement de la parole est le spectre, représenté au moyen des coefficients cepstraux du signal. Ces coefficients sont obtenus à partir du mod-

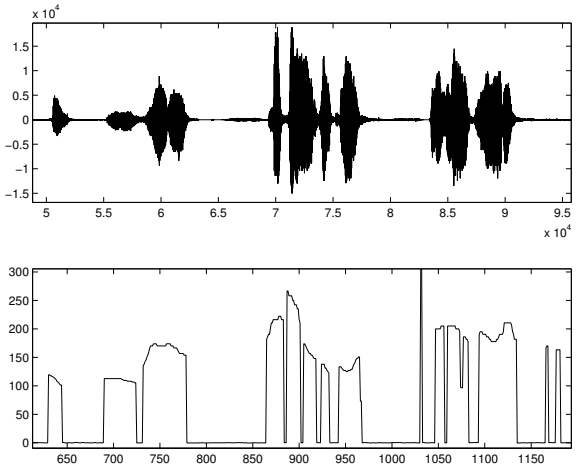


Figure F. 1.4: *Signal de Parole et le pitch correspondant.*

ule des coefficients de la Transformée de Fourier Discrète (TFD) du signal $s(n)$ sur lequel on applique successivement une fonction logarithme et une TFD inverse.

Une propriété importante des coefficients cepstraux est que le cepstre d'un signal filtré linéairement est égal à la somme du cepstre du signal d'entrée et du cepstre du filtre linéaire. De plus, si le filtre linéaire est également considéré comme invariant dans le temps, le cepstre du signal d'entrée peut être retrouvé par soustraction de la moyenne temporelle du cepstre du signal de sortie. Cette propriété est couramment employée pour éliminer les effets du canal de transmission. Une autre propriété fondamentale de l'analyse cepstrale est la décorrélation entre les coefficients issus de l'analyse. Cette propriété est très appréciable pour un système de VL et c'est principalement pour cette raison que l'analyse cepstrale est couramment employées.

1.4 Modèles Graphiques

Dans la section précédente, nous avons décrit différentes sources d'information que l'on peut obtenir à partir du signal de parole. A partir d'une combinaison appropriée de ces informations, il convient de construire un modéle qui rende le système de VL plus robuste. Les modèles graphiques sont un outil très puissant pour combiner différentes variables tout en préservant leurs dépendances conditionnelles. Dans cette section, nous nous attacherons à décrire et à définir de manière précise les modèles graphiques.

A l'origine, ce type de modèles a été développé indépendamment dans plusieurs domaines scientifiques distincts. En physique statistique, par exemple, leur origine peut être trouvée dans le travail de Gibbs [Gibbs, 1902]. Dans ce domaine, l'objectif initial était d'étudier un grand système de particules interagissant les unes avec les autres. Habituellement, on suppose que les particules agissent simplement avec les particules voisines. Pour modéliser les relations entre voisins dans un tel système, Gibbs a développé un modèle graphique particulier appelé distribution de Gibbs et défini par :

$$p(x) = \frac{1}{Z_T} e^{\frac{-E(x)}{T}},$$

où, T est la température du système, $E(x)$ l'énergie totale du système dans l'état x, et Z_T est une constante de normalisation. La distribution de Gibbs correspond à ce que l'on désigne comme un modèle graphique non dirigé car il n'y a aucune relation de hiérarchie entre voisins.

Les modèles graphiques ont été également développés dans le domaine de la génétique. Contrairement aux modèles non dirigés, les modèles conçus par les généticiens comme Wright [Wright, 1921] établissent une hiérarchie bien définie de relations entre les variables.

Wright a utilisé les modèles graphiques pour étudier les propriétés d'héridité des espèces. Notamment, il a introduit la notion de graphe pour modéliser les relations directes et unidirectionnelles entre variables. Ces relations sont représentées symboliquement par des flèches se déplaçant du parent à l'enfant. C'est probablement de ce domaine que provient la notation utilisée pour désigner les variables et leurs relations dans les modèles graphiques dirigés.

Ces idées d'analyse de chemin ont été reprises plus tard dans le domaine de l'économie et des sciences sociales [Wold, 1960; Blalock, 1971] notamment dans les travaux de Bartlett [Bartlett, 1935] qui décrit la notion d'interaction au sein d'une table de contingences.

Modèles Dirigés

La théorie des graphes est une partie de la théorie des ensembles qui traite des relations binaires d'un ensemble dénombrable avec lui-même. Un graphe G est dirigé si tous les arcs représentés sont dirigés. S'il existe une flèche ou arc qui pointe du nœud X_i vers le nœud X_j, X_i est appelé le parent de X_j et X_j l'enfant ou le fils de X_i. L'ensemble des parents de X_j est noté $Pa(X_j)$. Par exemple, dans la Figure ci-dessous le nœud a est le parent du nœud b et l'ensemble des parents de d est $Pa(d) = \{a, b, c\}$.

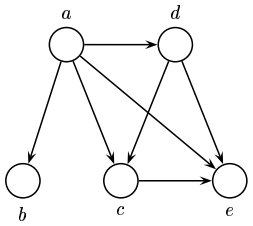


Figure F. 1.5: *Exemple d'un graphe dirigé illustrant les relations parents-enfants.*

Graphe Acyclique Dirigé

On appelle cycle tout chemin d'au moins deux nœuds reliés entre eux par des flèches et pour lequel le premier et le dernier nœud sont identiques. Une classe très importante de graphes dirigés est la classe des graphes ne présentant aucun cycle. On appelle Graphes Acycliques Dirigés (GAD) de tels graphes. Les GAD sont la base des modèles probabilistes appelés Réseaux Bayesiens.

Graphe Moral

Le graphe moral G^m d'un graphe dirigé G est défini comme le graphe non dirigé avec les mêmes nœuds que G où deux nœuds X_i et X_j sont reliés dans G^m si et seulement si X_i et X_j ont un enfants en commun dans G. Dans la pratique, le graphe moral est obtenu à partir du graphe original par "mariage" des parents ayant un enfant en commun et par suppression des flèches. On construit ainsi un graphe non dirigé à partir d'un graphe non dirigé.

Graphe Triangulé

Dans un graphe non dirigé, la terminologie est différente de celle employée pour les graphes dirigés. Ainsi, les liens entre les nœuds sont appelés arcs au lieu de flèches, et on désigne par le terme de boucle un chemin fermé qui correspond à un cycle dans un graphe dirigé. On appelle corde d'une boucle tout arc qui joint deux nœuds non voisins de cette boucle et qui n'est donc pas lui-même un arc de cette boucle. Un graphe triangulé est défini comme un graphe non dirigé qui a une corde dans chaque boucle de longueur n ≥ 4. La Figure suivante donne un exemple de graphe triangulé. La boucle $\{c, a, d, b\}$ a une corde $\{a, b\}$ qui rend cette boucle de longueur inférieure à 4.

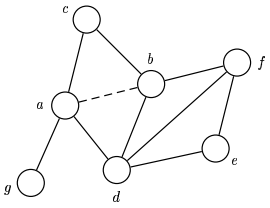


Figure F. 1.6: *Exemple de graphe triangulé. La corde* $\{a, b\}$ *change le cycle* $\{c, a, b, d\}$ *en deux cycles de longueur 3.*

Les concepts que nous venons d'introduire et le langage employé nous sont utiles pour décrire de manière rigoureuse les techniques qui sont présentées dans les prochains sections.

1.5 Réseaux Bayesiens

Les modèles graphiques sont des outils théoriques très puissants. Rappelons cependant que l'objectif recherché dans ce travail est de modéliser la parole à l'aide de réseaux bayesiens (RB). Un RB est un modèle graphique qui représente les indépendances conditionnelles entre un ensemble de variables aléatoires [Pearl, 1988]. Plus précisément, un RB est un couple (G, DPC) constitué par une structure G et un ensemble de distributions de probabilités conditionnelles. G est un graphe acyclique dirigé (GAD) et un ensemble de distributions de probabilité conditionnelles DPC caractérisant de façon quantitative chaque nœud de G. Pour illustrer de manière simple notre propos, considérons trois variables aléatoires A, B et C. D'après la théorie des probabilités, leur probabilité jointe s'écrit comme le produit des probabilités conditionnelles suivantes :

$$P(A, B, C) = P(A)\ P(B|A)\ P(C|A, B).$$

Si A est indépendante de B, l'équation précédente peut être écrite comme suit :

$$P(A, B, C) = P(A)\ P(B)\ P(C|A, B). \tag{1.1}$$

L'équation précédente peut être représentée par un graphe comme cela est illustré dans la Figure suivante. Chaque nœud correspond à une variable. Chaque flèche représente une dépendance entre les variables et est associée à la densité de probalité du fils sachant le parent. Un RB est juste une manière graphique de représenter les indépendances conditionnelles entre plusieurs variables. La structure reflète la factorisation de la distribution jointe.

Pour un ensemble de variables données $X = \{X_1, \ldots, X_N\}$, les relations d'indépendance conditionnelle induisent une factorisation de la fonction de distribution jointe $P(X)$ exprimée comme suit :

$$P(X) = \prod_{i=1}^{N} P(X_i|Pa(X_i)),$$

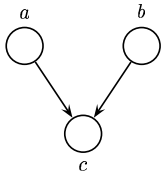


Figure F. 1.7: *Représentation graphique de l'équation 1 avec des Réseaux Bayesiens.*

Indépendance Conditionnelle

Soit un chemin orienté constitué d'une suite des nœuds $\{X_i, \ldots, X_j\}$ tel qu'un nœud donné X_k et le suivant X_{k+1} sont reliés par une flèche (avec $k \in [i; j-1]$). Soit X_k un nœud arbitraire avec $k \in [i+1; j]$. On appelle nœud non-descendant, par rapport à X_k tout nœud X_m tel que $m < k$.

Chacune des variables est conditionnellement indépendante de ses non-descendants étant donnés ses parents.

Un concept fondamental pour les modèles dirigés est la **d-séparation**, où **d** signifie "direct". Ce concept est très important pour la représentation des RBs. On dit que l'ensemble de variables C d-sépare les ensembles de variables A et B si pour chaque chemin entre A et B, il existe une variable d telle que :

(i) si $d \notin C$, alors d n'a que des flèches convergentes vers elle ou,

(ii) si $d \in C$, alors d a au moins une flèche divergente.

Deux ensembles de variables A et B disjoints sont conditionnellement indépendants si et seulement si il existe un ensemble C (éventuellement vide) qui *d-sépare* A et B.

Graphes équivalents

Un problème proche de celui de l'apprentissage de la structure est celui de savoir si deux structures sont différentes. Deux modèles graphiques sont équivalents s'ils représentent le même ensemble d'indépendances conditionnelles [Verma and Pearl, 1991]. Dans une graphe dirigé, la même fonction de probabilité conditionnelle peut être représentée avec plusieurs graphes. Afin de définir l'équivalence entre deux RBs, il faut en premier lieu définir la notion de V**-structure**. Considérons trois nœuds $\{a, b, c\}$ d'un RB. Ils forment une structure aux flèches convergentes, si a et c sont reliés par la flèche qui va de a vers c et b et c par la flèche qui va de b vers c comme cela est illustré dans la Figure F.7 plus haut. En raison de sa forme généralement en V, on appelle alors V**-structure** ce type de structure. On dit que deux RBs sont équivalents si les deux graphes ont la même structure non dirigée et les mêmes V**-structures**.

1.5.1 Réprésentation des lois de probabilités à l'aide de RB

On considère trois types de distribution de données qui peuvent être réprésentés par des RB : la loi Multinomiale, la loi Multinormale et une distribution plus générale construite avec un modèle de mélange de gaussiennes (MMG). Dans un RB multinomial, toutes les variables $\{x_i\}$ sont discrètes et la fonction de probabilité conditionnelle associée à chaque variable $\{x_i\}$ est une fonction multinomiale. Ce type de fonctions de probabilité est défini de manière numérique ou paramétrique à l'aide des Tableaux de Probabilité Conditionnelle (**TPC**). Ces tableaux indiquent les probabilités pour chacune des combinaisons possible des valeurs prises par les variables. Par exemple, les paramètres (TPC) pour le graphe de la Figure F.7 pourraient être ceux indiqués dans les tableaux suivants en supposant que toutes les variables sont binaires $X = \{x, \neg x\}$:

a	$p(a)$
0	0.25
1	0.75

b	$p(b)$
0	0.2
1	0.8

a	b	$p(c\|a,b)$
0	0	0.25
0	1	0.50
1	0	0.75
1	1	0.50

RB Multinormal

Dans un RB gaussien ou normal, toutes les variables sont modélisées par une distribution normale $\mathcal{N}(x;\mu,\sigma)$. Ce genre de RB s'appellent également Réseau Gaussien (RG) [Shachter and Kenley, 1989] laissant l'appelation RB pour des réseaux de variables discrètes. La loi de distribution normale est donnée par l'équation suivante :

$$f_{\mathcal{N}}(x) \backsim \mathcal{N}(x;\mu,\Sigma) = \frac{1}{(2\pi)^{-\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}.$$

où μ est le vecteur moyen de dimension d, Σ est la matrice de covariance, $|\Sigma|$ le déterminant de Σ et $(x-\mu)^T$ est la transposée de $(x-\mu)$. La fonction de densité pour chaque facteur dans l'équation précédente est définie par un produit des fonctions de probabilité conditionnelle [Shachter and Kenley, 1989] comme suit :

$$f(x_i|Pa(x_i)) \backsim \mathcal{N}(x;\mu_i + \sum_{j=1}^{i-1}(\beta_{i,j}(x_j-\mu_j),\nu_i),$$

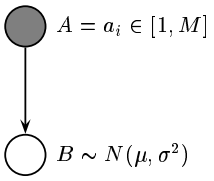
où $\beta_{i,j}$ est le coefficient de régression entre x_i et ses parents x_j.

RB et Modèle de Mélange de Gaussiennes

Un modèle de mélange de gaussiennes (MMG) est défini comme une combinaison pondérée de densités gaussiennes. La densité conditionnelle pour un vecteur **x** représenté par un mélange de M composantes est définie par $\Lambda = \{w,\mu,\Sigma\}$ telle que :

$$p(x|\Lambda) = \sum_{i=1}^{M} w_i f_{\mathcal{N}}(x),$$

où, $\sum_{i=1}^{M} w_i = 1$ et $0 \leq w_i \leq 1 \ \forall i$. La même distribution de probabilité peut être représentée à l'aide d'un RB comme cela est illustré dans la Figure suivante.



$$p(B=b) = \sum_{i=1}^{M} p(B=b|A=a_i)p(A=a_i).$$

Figure F. 1.8: *Réprésentation d'un MMG avec un RB.*

Dans la Figure F.8, le premier nœud A, représente une variable discrète avec M états qui vérifie $\sum_{a_i} p(a=a_i) = 1$ et $0 \leq P(A=a_i)$.

Le deuxième nœud B représente une variable qui suit une distribution gaussienne conditionnée à la valeur prise par la première variable $A = a_i$. Avec les indépendances conditionnelles réprésentées dans la structure, la probabilité conditionnelle s'écrit selon l'équation indiquée dans la même Figure. Cette même structure représente un MMG avec les paramètres suivants :

$$\begin{cases} p(A = a_i) = w_i, \\ p(B = b|A = a_i) = \mathcal{N}(b; \mu_i, \Sigma_i). \end{cases}$$

1.5.2 Réseaux Bayesiens Dynamiques

Le temps est une des variables les plus importantes dans presque tous les évènements liés à des processus réels. L'évolution des variables dans le temps peut également être représentée avec des Réseaux Bayesiens Dynamiques (RBD). Les RBD permettent de décrire un système qui change ou évolue avec le temps en utilisant le formalisme des RBs. Les RBDs sont définis en tant que cas particulier des RBs. Tous les nœuds, arcs et probabilités qui forment un RBD ont les mêmes interprétations statistiques que pour un système basé sur un RB classique. Les états d'un RBD satisfont les conditions markoviennes.

Un RBD se compose de fonctions de distribution de probabilité pour la séquence des variables cachées $H = \{h_0, \ldots, h_{T-1}\}$ et des variables observées $O = \{o_0, \ldots, o_{T-1}\}$, où T est l'indice temporel. Si toutes les variables cachées et observées sont intégrées au sein d'une même variable $X = \{h, o\}$, la sémantique d'un RBD peut être définie en déroulant un RB jusqu'à ce qu'ayant finisse avec toutes les tranches de temps Figure F.9. La distribution résultante de cette modélisation peut être écrite comme suit :

$$P(x_{1:T}) = \prod_{t=1}^{T} \prod_{i=1}^{N} P(x_i(t)|Pa(x_i(t))),$$

où N est le nombre de variables dans chacune de tranches de temps.

Un des exemples les plus représentatifs de ce type de modèles est celui des Modèles de Markov Cachés (Hidden Markov Models, HMM). La particularité d'un HMM par rapport à un RBD est que, dans un HMM, l'espace d'état se compose d'une seule variable aléatoire X_t alors que dans un RBD les états cachés sont représentés à travers un ensemble de variables aléatoires $X_1, \ldots, X_t$.

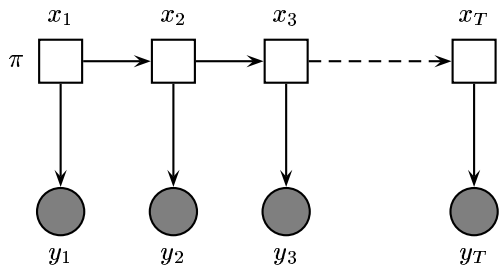


Figure F. 1.9: *Un HMM représenté en tant qu'un RBD.*

Dans cette section, nous avons présenté les principes des RBs et des RBD qui laissent déjà présentir la possibilité de combiner les différents sources d'information au sein d'un unique modèle statistique. Les prochaines sections présenteront les problèmes de base auquel on est confrontés dans les modèles graphiques, à savoir l'inférence et l'apprentissage. Nous développeront le concept de modélisation dans la chaîne d'un système de VL.

1.6 Inférence dans les RB

L'absence de flèches dans les RBs indique des indépendances conditionnelles qui peuvent être exploitées pour développer de meilleurs algorithmes de calcul des probabilités marginales et conditionnelles. Il y a deux problèmes de recherches principaux dans le raisonnement probabiliste utilisant des RB. Le premier problème est l'étude de l'inférence [Murphy, 2002]. L'inférence dans un RB implique, dans une structure connue, le calcul de la probabilité marginale *a posteriori* de quelques variables sachant la valeur des variables observées. Le deuxième problème est celui de l'apprentissage de la structure qui représente les indépendances conditionnelles entre les variables.

Cette section est consacrée aux techniques d'inférence basiques [Murphy, 2002]. Le problème abordé consiste à savoir comment évoluent les probabilités conditionnelles d'une ou plusieurs variables X dans un réseaux sachant la valeur des variables observées $Y = y$. Plus précisément, nous cherchons la probabilité $P(X|Y = y)$, où $Y = y$ est l'observation, couramment appelée *évidence*. Ces techniques sont importantes car elles servent de base pour le calcul des paramètres d'un réseau avec des variables non observées appelées *variables cachées*.

La manière la plus simple d'évaluer la probabilité cherchée est de marginaliser la fonction de probabilité jointe. Cependant, ce procédé est inefficace car quelques calculs peuvent être faits plusieurs fois. Une des techniques les plus directes est appelé *élimination de variables*. Supposons qu'on souhaite calculer la probabilité d'une variable aléatoire x_i. L'élimination de variables consiste alors à établir un ordre dans les variables de telle façon que x_i soit la dernière. A chacune des étapes, on élimine une des variables distinctes de x_i en combinant tous les facteurs où elle est présente et en la marginalisant. Le résultat final est un potentiel qui est proportionnel à la probabilité cherchée. Par exemple, le graphe de la Figure ci-dessous réprésent la probabilité jointe suivante :

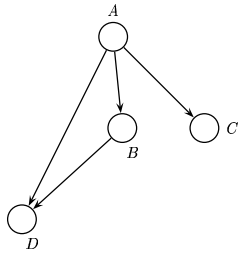


Figure F. 1.10: *RB utilisé pour illustrer la techniques Élimination de Variables pour le calcul de l'inférence.*

$$P(A, B, C, D) = P(A)\ P(B|A)\ P(C|A)\ P(D|A, B).$$

Pour illustrer cette technique, on calcule la probabilité de la variable A. $P(A)$ peut être alors évaluée en marginalisant la probabilité jointe par rapport aux variables $\{B, C, D\}$ comme suit :

$$\begin{aligned} P(A) &= \sum_{B,C,D} P(A, B, C, D) \\ &= \sum_{B,C,D} P(A)\ P(B|A)\ P(D|A, B)\ P(C|A). \end{aligned} \tag{1.2}$$

Sur cette dernière équation, on remarque que les facteurs de la probabilité jointe ne dépendent que d'un nombre limité de variables et par conséquent la marginalisation peut être réécrite de la manière suivante :

$$P(A) = P(A) \sum_B P(B|A) \sum_D P(D|A, B) \sum_C P(C|A).$$

En général, un facteur de la forme $\sum_X P(X|Y)$ est égal à l'unité mais ce n'est pas toujours le cas. Par exemple, dès que la variable X est observée et qu'elle prend une valeur, alors la somme est différente de l'unité.

Passage de Messages en Polyarbres

L'élimination de variable est une technique qui réduit le nombre d'opérations nécessaires pour calculer la probabilité cherchée. Cependant, elle demande encore beaucoup des calculs. Il existe d'autres techniques plus efficaces qui tirent avantage des indépendances conditionnelles existant dans une structure en forme d'arbre [Pearl, 1988]. Même si, dans des applications réelles, les problèmes n'induisent pas toujours une structure en forme d'arbre, les techniques appliquées à ce type de graphes sont la base pour des structures plus générales.

Le passage de message dans un polyarbre (une structure en arbre ou plusieurs racines sont permis) est une technique qui s'appuie sur la structure d'arbre du réseau et sur la d-séparation entre ses variables. En effet, dans un polyarbre, chaque nœud d-sépare ses descendants de ses non-descendants. De la même façon, chaque variable est d-séparée de ses frères conditionnellement à ses parents et aussi à chacun des parents de ses fils conditionnellement à ses fils.

Dans un polyarbre, il n'y a qu'un seul chemin entre deux variables. Chacun des nœuds coupe alors la structure en deux polyarbres complètement séparés. De la même façon, on coupe l'évidence en deux. On distingue alors d'une part l'évidence qui arrive à travers les parents du nœuds en question et d'autre part celle qui arrive par l'intermédiaire de ses fils. Etant donné que ce même nœud d-sépare les deux structures mentionnées, la probabilité conditionnelle cherchée est proportionnelle au produit des deux probabilités, chacune liée à une de ces structure. Ces termes qui sont appelés λ et π peuvent être vues comme des messages envoyés par chacune des structures au nœud qui les sépare[1]. Du fait de la d-séparation, ces messages peuvent être décomposés en parties correspondant à chaque variable connectée au nœud en question. Dans le réseau, chaque variable reçoit un message de chacun de ses parentes et de chacun de ses fils. Une fois que tous les messages ont été reçus, la variable peut envoyer, elle aussi, un message à ses voisins. A la fin de ce processus, toutes les variable ont reçues l'information provenant de l'évidence.

Arbre de Jonction

Lorsque la structure originale du réseau n'est pas un arbre, une des techniques les plus utilisées consiste à transformer la structure initiale pour obtenir un arbre non dirigé. Une fois que le modèle graphique non dirigé est obtenu, des calculs d'inférence sont réalisés en utilisant le formalisme des graphes non dirigés. En général, le graphe final est un arbre constitué de cliques. On rappelle qu'une clique est un sous-graphe complètement connecté.

La première étape de conversion d'un graphe dirigé en un graphe non dirigé est la *moralisation*. Comme cela a été mentionné avant, la moralisation dans un RB est le processus qui consiste à "marier" les parents ayant un enfant commun et ensuite à supprimer les directions des flèches pour obtenir un graphe non dirigé. Si besoin, la deuxième étape de construction d'un arbre de jonction est la triangulation. Une fois qu'un graphe est triangulé, il est possible d'arranger les cliques du graphe dans une structure qu'on appelle *arbre de jonction*. Dans un arbre de jonction, si un nœud appartient à deux cliques quelconques de l'arbre, alors il appartient également à toutes les cliques qui se trouvent sur le chemin entre ces deux cliques. Cette propriété permet de calculer l'inférence en se basant sur des calculs locaux.

Une fois que l'arbre de jonction est construit, un potentiel est attribué à chacune des cliques. Pour chaque clique, ce potentiel est en relation directe avec les probabilités conditionnelles des variables de

[1] λ et π sont équivalents aux α et β dans un HMM.

celle-ci. Ces potentiels jouent un rôle similaire à celui des messages dans l'algorithme de passage de messages dans un polyarbre décrit dans la section précèdente. Les potentiels sont actualisés au cours d'étapes dites de propagation. Dans un premier temps, chaque clique collecte les messages en provenance des cliques voisines et puis elle distribue à son tours son prope message. Si l'initialisation est correcte, ces deux étapes permettent de trouver un équilibre dans l'arbre.

Etant donnée la complexité des méthodes exactes d'inférence dans les RB avec beaucoup de variables, il est nécessaire d'avoir recours à des algorithmes de calcul approché. Une première technique de calcul, nommée "Loopy Belief Propagation", utilise des techniques exactes comme le passage de message en polyarbres dans des réseaux dont la structure comporte des cycles. Les autres techniques considérées sont plus rigoureuses et se basent sur des méthodes d'échantillonnage comme celle de Gibbs.

1.7 Apprentissage des Réseaux Bayesiens

L'apprentissage de RBs consiste à obtenir automatiquement la structure et/ou les paramètres à partir de l'information contenues dans les données disponibles. On distingue quatre variantes de ce problème :

- structure connue et base de données complète
- structure connue et données manquantes ou cachées
- structure inconnue et base de données complète
- structure inconnue et données manquantes ou cachées

La liste des problèmes ci-dessus est présentée dans un ordre croissant de difficulté.

L'apprentissage des paramètres d'un réseau dont on connaît la structure est réalisée en utilisant des techniques classiques comme le maximum de vraisemblance (Maximum Likelihood, ML). Par contre, dès que la base de données est incomplète, la meilleure option est l'algorithme Expectation Maximization (EM). Cet algorithme utilise dans une première étape des techniques d'inférence pour calculer les paramètres manquants du réseau. Dans la deuxième étape, il effectue une maximisation de tous les paramètres. Ces deux étapes sont répétées de façon itérative jusqu'à la convergence.

Le problème de l'apprentissage de la structure est plus complexe que celui de l'apprentissage des paramètres. On peut distinguer deux approches bien différentes. La première ajoute ou enlève des flèches dans la structure en fonction des résultats d'une recherche des indépendances conditionnelles entre les variables. Ce type d'algorithme, comme celui appelé PC [Pearl, 1988], est initialisé avec un réseau complètement connecté et au fur et à mesure qu'il trouve des indépendances, les flèches adéquates sont enlevées.

La deuxième approche mesure la qualité d'un réseau donné et choisit la structure qui donne le meilleur score. Cette technique nous confronte cependant à deux problèmes. En premier lieu, il faut déterminer une fonction qui mesure la qualité du réseau en fonction des données. La vraisemblance est un bon candidat, mais elle privilègie les structures ayant un grand nombre de connections ; c'est-à-dire qu'elle privilègie les réseaux avec plus de paramètres car la vraisemblance augmente avec le nombre de paramètres. Alors, à la valeur de la vraisemblance, peut être ajouté un facteur qui pénalise la complexité du réseau. Un exemple de fonction qui remplit ce rôle est le score BIC (Bayesian Information Criterion). Ce facteur supplémentaire est fonction du nombre de paramètres du réseau et de la quantité de données disponibles. Le deuxième problème dans ce type d'approche est de choisir la structure qui sera évaluée. On peut penser à évaluer toutes les structures possibles pour un réseau. Mais cette démarche n'est pas envisageable car le nombre de structures croît exponentiellement avec le nombre de variables. Par exemple, avec trois variables, il y a 25 structures possibles ; avec 5 variables, il y en a 29281. Le coût de calcul devient alors rapidement prohibitif. Il faut remarquer que parmi toutes les structures possibles, plusieurs sont équivalentes du point de vue Markovien et le problème pourrait donc être simplifié. Mais, même avec cette réduction, il reste

beaucoup de structures à évaluer. Pour surmonter ce problème, une réduction de l'espace de recherche peut être effectuée, notamment en introduisant des connaissances *a priori*. Une manière très répandue de réduire l'espace est d'introduire un ordre sur les variables concernées. Cet ordre, appelé *ordre ancestral*, indique les relations de filiation et de hiérarchie entre les variables. Un autre avantage à l'utilisation d'un tel ordre est qu'il permet d'éviter les cycles. Par exemple, les algorithmes de recherche de structure de type glouton s'initialisent à partir d'un réseau de départ spécifié puis ils construisent des structures voisines. Une structure voisine d'une structure donnée est une structure où une flèche a été soit ajoutée, soit changée d'orientation, ou soit enlevée. Pour chacune de ces structures, le score est ensuite calculé et on choisit comme structure de départ de l'itération suivante celle qui a le meilleur score.

1.8 Adaptation des modèles

Dans la chaîne du système de VL que l'on est en train de détailler, la modélisation à l'aide de RB peut poser, en pratique, des problèmes comme n'importe quelle autre modélisation. Une manière courante d'obtenir des modèles robustes est d'adapter les modèles utilisés. Ici, on entend par adaptation, le processus ou la technique par lesquels on réduit l'influence respective du manque de données disponibles et des différences entre les environnements d'apprentissage et de test.

Dans le domaine de la VL, il y a deux approches principales pour faire l'adaptation. La plus importante et celle qui donne les meilleurs résultats est l'adaptation Bayesienne (Maximum A Posteriori, MAP). Dans cette méthode, le modèle adapté est calculé en maximisant la probabilité a posteriori des paramètres conditionnellement aux données d'apprentissage disponibles. On utilise une loi *a priori* sur les paramètres du modèle. Cet *a priori* qui reflète les connaissances du modèle dont on dispose, est une des données les plus importantes dans cette technique. La loi *a priori* peut être choisie soit de manière empirique ou en se basant sur l'expérience ou les données disponibles, soit pour des convenances mathématiques. Les paramètres de la loi *a priori*, appelés hyper-paramètres, sont calculés soit de manière empirique à partir des connaissances de la vraie distribution, soit avec un ensemble de données représentatif de cette distribution. Dans le domaine de la VL, le deuxième choix est préférable. Les paramètres du modèle adapté sont en général une combinaison des paramètres de la loi *a priori* et de la distribution des données d'apprentissage. Par exemple, dans un MMG, la moyenne de chacune des composantes est la combinaison de la moyenne de la composante *a priori* correspondante et de la moyenne calculée avec les données d'apprentissage. Cette combinaison est pondérée par des facteurs qui donnent plus ou moins d'importance à la gaussienne initiale ou à celle dépendant des données d'apprentissage. Dans le cas extrême, l'adaptation ne se fait que pour les composantes pour lesquelles on dispose de données.

La deuxième approche consiste à effectuer une transformation linéaire sur les paramètres du modèle original pour obtenir le modèle adapté. Si les paramètres de régression utilisés pour l'adaptation sont estimés avec le maximum de vraisemblance (ML), cette technique est appelée MLLR (Maximum Likelihood Linear Regression). Dans cette approche, la quantité de données disponible joue un rôle très important. S'il y a très peu de données, une seule transformation est estimée pour tous les paramètres. Au fur et à mesure que la quantité de données augmente, il est alors possible de prendre en compte plusieurs transformations, chacune s'appliquant à un ensemble de paramètres déterminé.

1.8.1 Adpatation de Réseaux Bayesiens

En matière d'adaptation des RB, notre approche est basée sur la technique Bayesiennes pour laquelle on adapte les paramètres qui déterminent les relations d'indépendance conditionnelle entre deux variables. Deux configurations distinctes de dépendance conditionnelle ont été étudiées. Elles dépendent du type de variables concerné et nous amène à distinguer les relations entre deux variables discrètes et celles entre deux variables continues. Dans le cas de variables discrètes, pour des raisons mathématiques, on choisit

d'utiliser une distribution de Dirichlet comme loi *a priori*. Si chacun de paramètres d'un Tableaux de Probabilité Conditionnelle (**TPC**) est définit comme suit :

$$\theta_{ijk} = P(x_i = j|Pa(x_i) = k),$$

et la distribution de Dirichlet est :

$$P(\theta|\alpha) = \frac{1}{Z(\alpha)} \prod_{i=1}^{N} \theta_{ijk}^{\alpha_{ijk}-1},$$

les valeurs optimales pour θ_{ijk} de la distribution *a posteriori* après l'optimisation sous la contrainte suivante :

$$\sum_{j=1}^{N} \theta_{ijk} = 1, \tag{1.3}$$

sont comme suit :

$$\widehat{\theta}_{ijk}^{MAP} = \rho_{ik} \frac{N_{ijk}}{\sum_{j=1}^{N} N_{ijk}} + (1 - \rho_{ik}) \frac{\alpha_{ijk} - 1}{\sum_{j=1}^{N} \alpha_{ijk} - 1}.$$

où :

$$\rho_{ik} = \frac{\sum_{j=1}^{N} N_{ijk}}{\sum_{j=1}^{N} N_{ijk} + \alpha_{ijk} - 1},$$

et :

$$N_{ijk} = \sum_{t=1}^{T} \mathbf{1}_{(x_i(t)=j, Pa(x_i(t))=k)}.$$

Après les calculs d'optimisation, les paramètres des tableaux de probabilités adaptés sont estimés comme une combinaison linéaire des paramètres des tableaux de la distribution *a priori* et du comptage des configurations dans les données d'apprentissage.

Dans le cas de variables continues, la loi *a priori* choisie est une distribution Normale-Wishart, définit comme suit :

$$p(x, m_x, m_y, \mathbf{r}, \mathbf{B}|\tau, \mu, \alpha, \mathbf{u}) \propto |\mathbf{r}|^{\frac{\alpha-p}{2}} \exp -\frac{\tau}{2}(m_y + \mathbf{B}(x - m_x) - \mu)^t \mathbf{r}(m_y + \mathbf{B}(x - m_x) - \mu) \exp -\frac{1}{2} tr(\mathbf{ur}).$$

Les paramètres de la matrice de régression qui modélise les relations d'indépendance conditionnelle sont optimisés et le résultat est le suivant :

$$\begin{aligned} \widehat{\mathbf{B}} \quad = \quad & \beta \left(\sum_k (y^k - m_y)(x^k - m_x)^t \right) \left(\sum_k (x^k - m_x)(x^k - m_x)^t \right)^{-1} + \\ & (1 - \beta) \left(\sum_k (\mu - m_y)(x^k - m_x)^t \right) \left(\sum_k (x^k - m_x)(x^k - m_x)^t \right)^{-1}, \end{aligned}$$

où :

$$\beta = \frac{1}{1 + \tau},$$

et τ est un paramètre de la distribution Normale-Wishart. Le paramètre optimal obtenu est, une fois de plus, une combinaison linéaire entre les paramètres de la distribution gaussienne *a priori* et de ceux estimés à partir des données d'apprentissage.

1.9 Vérification du Locuteur à l'aide de Réseaux Bayesiens

Les sources d'information choisies pour construire le système de VL sont les coefficients cepstraux, le résiduel de l'analyse en PL (qui modélise la source de production de la parole) et enfin le pitch et l'énergie (en ce qui concerne la prosodie). Une fois que les paramètres du signal acoustique ont été correctement établis, la deuxième étape consiste à déterminer les relations d'indépendance conditionnelle entre ces paramètres. On utilise deux algorithmes différents pour retrouver ces relations. Le premier algorithme, inspiré de l'algorithme K2 [Cooper and Herskovits, 1992] utilise une recherche de structure de type glouton avec un ordre préétabli et une fonction de score BIC. Le deuxième algorithme, quant à lui, emploie une mesure de qualité basée sur l'approche MDL (pour Minimum Description Length) [Sigelle, 2003] et n'utilise que des variables discrètes.

Figure F. 1.11: *Structures qui réprésentent l'énergie (E), le pitch (F_0), le spectre (SLPCC) et la source de production (RLPCC) obtenus avec les algorithmes K2 à gauche et MDL à droite.*

Les structures obtenues ont une interprétation physique assez intéressante. Comme la première approche n'utilise que des variables continues, la structure reflète l'aspect de production de la parole. Une relation directe entre l'énergie et le fréquence fondamentale est trouvé ainsi qu'entre le spectre et le résiduel de l'analyse en PL. Par contre la structure obtenue avec des variables discrètes reflète une relation des zones de l'espace acoustique obténue en quantifiant les variables continues.

Une fois trouvés les relations d'indépendance conditionnelle, on a décidé de modéliser chacune de variables avec un MMG. Maintenant chacune des sources d'information est modélisée avec deux variables, une variable discrète qui représente les poids dans le mélange et une variables continue qui représente une distribution gaussienne. Ce choix nous a permis de spécifier deux types de relation entre les variables et d'en envisager une autre. On peut, soit relier les variables discrètes, soit les variables continues, soit toutes les deux. Si le premier choix est utilisé une dépendance est établie entre les variables qui décident les gaussiennes utilisées pour modéliser chacune des observations. Le deuxième choix met en relation de dépendance les variables continues directement entre elles. Très peu d'influence de la structure sur les résultats a été trouvé, spécialement avec une relation entre les variables discrètes, voir Figure F.12.

Dans les résultats mentionnés, seule une adaptation classique des gaussiennes a été utilisée. Dans les techniques d'adaptation proposées dans le cas de relations entre variables discrètes, on peut faire deux remarques. D'une part, chacun des tableaux de probabilités est une matrice stochastique et, d'autre part, l'adaptation est faite en combinant des colonnes de ces matrices. Comme les matrices sont stochastiques, alors chaque colonne est une distribution de probabilités. En modifiant les valeurs de ces distributions, on modifie également les relations d'indépendance entre les variables concernées. Si, de plus, on peut mesurer une distance entre ces deux distributions de probabilité, on peut également mesurer les variations des indépendances entre les variables.

Dans nos expériences, on propose de réaliser l'adaptation en utilisant ce type de mesure comme facteur de pondération dans la combinaison des modèles. Les premiers résultats ont été obtenus avec des variables

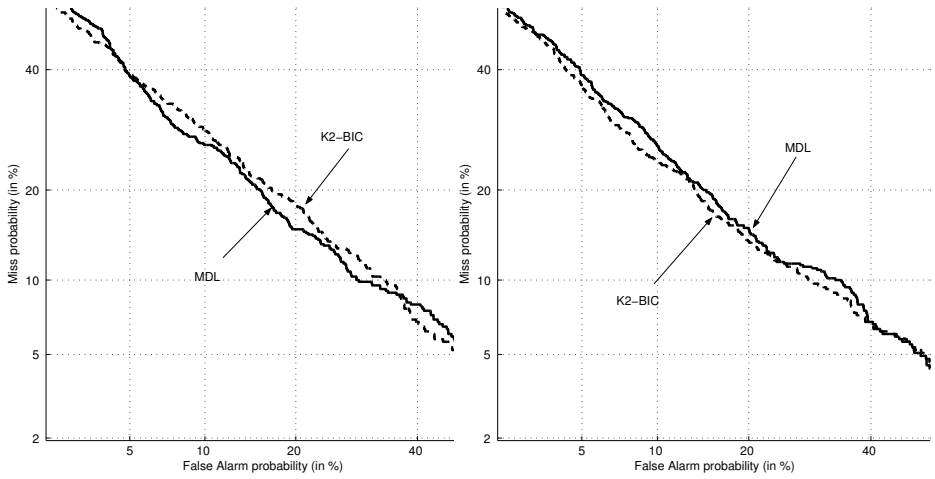


Figure F. 1.12: *Courbes DET pour les systèmes basés sur les structures obtenues avec les algorithmes K2 et MDL.*

discrètes. Chacune de ces variables a été discrétisée en utilisant une quantification vectorielle. Les résultats montrent de façon très claire l'influence de la distance entre les distributions de probabilités qu'on utilise (Kullback-Leiber, Aitchison et une valeur fixe), Figure F.13.

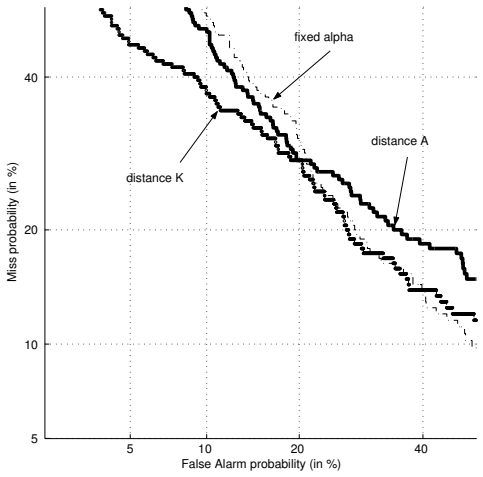


Figure F. 1.13: *Courbes DET pour les systèmes basés sur les structures obtenues avec les algorithmes K2 et MDL.*

Les dernières expériences réalisées au cours de notre travail appliquent l'adaptation pour des RB dont les variables sont modélisées à l'aide de MMG. D'abord, si les dépendences sont mis entre les variables discrètes on a des structures comme celles montrés dans la Figure suivante.

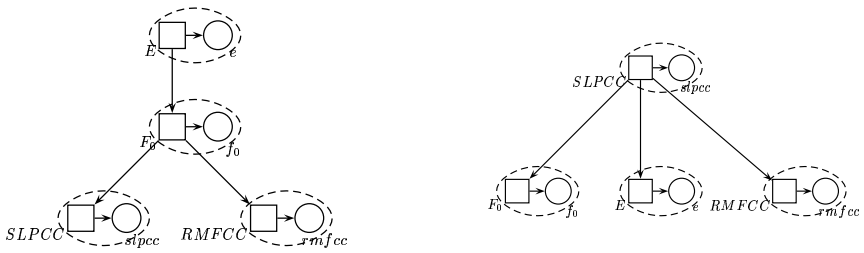


Figure F. 1.14: *Structures avec des dépendances entre les variables discrètes.*

Les résultats obtenues avec ces structures et un facteur d'adaptation ρ_{ik} fixé sont montrées dans la Figure F.15. Pour cet expérience on a utilisé 8 composants pour $slpcc$ et $mfcc$, 3 pour f_0 et 2 pour e. Les coefficients $slpcc$ et $mfcc$ ont été normalisés avec une Gaussianisation de la distribution des variables [Pelecanos and Sridharan, 2001].

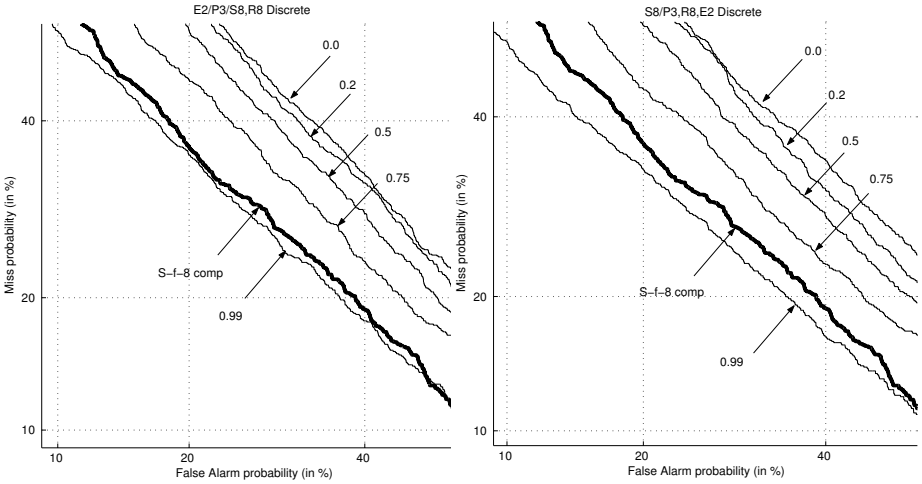


Figure F. 1.15: *Courbes DET pour les systèmes K2 et MDL avec des relations discrètes adaptés.*

Et finalement, si les dépendances sont mis entre les variables continuous on a des structures comme celles montrés dans la Figure suivante.

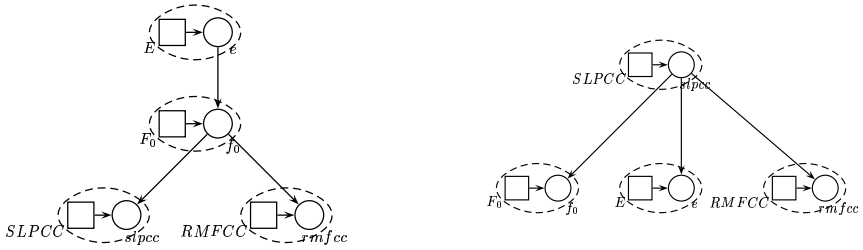


Figure F. 1.16: *Structures avec des dépendances entre les variables discrètes.*

Pour cet expérience on a utilisé 8 composants pour $slpcc$ et $mfcc$, 5 pour f_0 et 4 pour e. Les coefficients $slpcc$ et $mfcc$ ont été centrés est réduites. Les résultats obtenues avec ces structures et un facteur d'adaptation ρ_{ik} fixé sont montrées dans la Figure F.17.

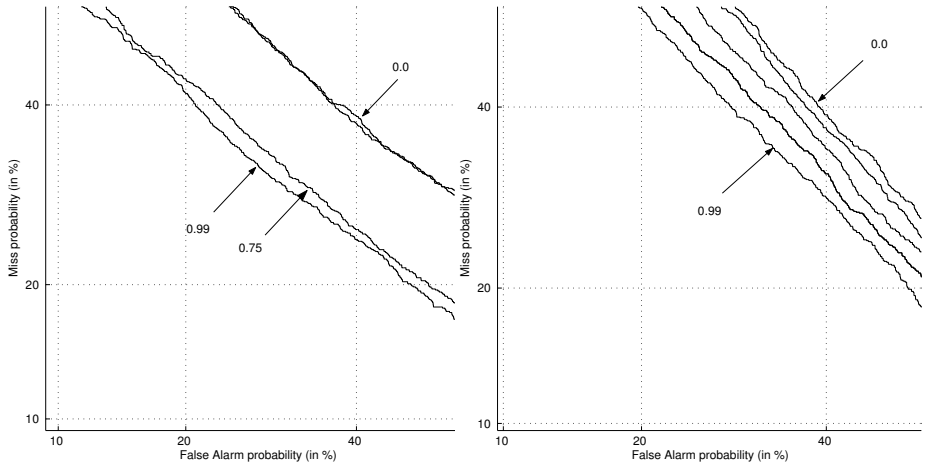


Figure F. 1.17: *Courbes DET pour les systèmes K2 et MDL avec des relations continuous adaptés.*

Ces résultats ne montrent pas de grands changements par rapport à l'adaptation dans le cas discret. En revanche, dans le domaine continue, l'adaptation des relations permet d'améliorer les résultats.

1.10 Conclusions

Au cours de ce travail de thèse, nous avons proposé de tirer parti du formalisme des réseaux bayesiens pour réaliser un système de VL. Nous avons proposé les RB comme un outil de modélisation des dépendances entre les attributs du signal de parole. Nous avons également développé une nouvelle approche pour combiner ces différentes caractéristiques porteuses d'information sur l'identité d'un locuteur. La combinaison des informations spectrales et prosodiques est effectuée au niveau des données et non au niveau des scores. Les dépendences apprises nous ont conduit à l'obtention des structures qui reflètent les realtions physyques entre ces attributs.

Une nouvelle technique d'adaptation pour la VL a aussi été présentée. Dans le système que nous proposons, l'adaptation des TPCs et des matrices de régression d'un Réseau Bayesien est faite par combinaison de leurs valeurs respectives dans les modèles du monde et dans les données d'apprentissage. Les résultats montrent que les performances du système issu de cette adaptation sont meilleures que celles d'un système où seules les moyennes des gaussiennes sont adaptées.

A l'issue de nos recherches, l'intérêt d'employer les Réseaux Bayesiens dans les systèmes de VL est clairement montré. Les résultats obtenus sont encourageants dans le cadre de la VL même s'ils ne sont pas comparables aux performances des systèmes qui représent l'état de l'art. En effect, nous nous sommes intéressés prioritairement à l'étude du potential des RB comme un outil statistique pour modéliser plus fidèlement les signal de parole.

Les modèles ainsi obtenues sont susceptibles de fournir de meilleures performances, dans des conditiones d'apprentissage plus optimales. Par conséquent, une poursuite des travaux de recherches dans ce domaine nous semble particulièrement judicieux et pertinent dans l'objectif de l'amélioration des performances des systèmes de VL.

Chapter 1

General Introduction

Speech Processing (SP) is a field with many applications. Analysis, Synthesis, Coding and Recognition are the basic areas of SP. Speech Recognition can be divided into three main sub-areas: Language Identification, Speech Recognition and Speaker Recognition (SR). Likewise SR can itself be divided into Speaker Identification (SI) and Speaker Verification (SV). In SI there is no a priori claimed identity and we would like to find out this identify among a group of persons. Then, the answer in such a task is the speaker's identity or the group to whom the speaker belongs. Or, in the open-set case the output could be that the speaker is unknown to the system because the actual speaker has no model since he/she comes from an unknown speaker group. SV systems, the main topic of this work, verifies, accept or reject, the identity claimed by a given speaker using the available information. This information, in most of the cases, is just the speech signal, as is the case in a telephone call. Such a system has to deal with two different events. The first occurs when the speech corresponds to the claimed identity. The second event occurs when the claimed identity does not correspond to the observed speech. In the first case the person who speaks is called a client and in the second cases is called an impostor. Therefore, the system can just accept a person, because it decides that he/she is a client, or reject him/her because it decides that he/she is an impostor.

Performance of actual SV systems is still different from that of humans. The first difficulty in a real application is to deal with non-cooperative speakers. If the speaker does not collaborate the signal quality could be degraded. In addition the system has to be prepared to confront some unexpected circumstances. Unlike of non-cooperative speaker, the speakers who want to be identified can be imposed to speak a defined sequence of words, for example a password. This utterance or flow of words can be defined each time that the speaker uses the system (a random sequence) or can be fixed before the utilization of the system. This constraint is used to class SV systems into text dependent and text-independent. Our work is addressed to text-independent SV system. The second complication in real applications lies in dealing with non-controlled scenarios. The system has to deal with different background noises and/or different communication channels. SV systems have to be prepared to come across with those difficulties. These constraints are reflected in all the processing steps before making the final decision.

An important problem in SV is to find the right feature set. One difference between systems and humans is the amount of used information. Speech is the most natural and commonly manner used by humans to communicate to each other. Consequently it is a really rich source of information. Just a few seconds of speech convey a very large amount of information. The most important one among those informations is the message. Speech serves as a medium in carrying essentially the main or most important ideas that someone is trying to tell to someone else. But, in addition to that message another very much important characteristic information is also present into the speech signal. This important characteristic is the identity of the speaker.

The message as well as the identity of the speaker are coded in several levels of abstraction. Every person has a different voice, a different way of speak, a different rhythm of speaking, a different tone of voice, some favorite words, etc. From acoustic to linguistic and paralinguistic levels speech helps to coding

certain intentions expressed into the utterance which can be unique to a special, and unique speaker.

In the acoustical level the spectrum could help someone or the system to identify his/her interlocutor (in the case of humans), but humans do not just use acoustical information in a normal communication. Usually each person uses all the information mentioned in the last paragraph to identify a person. They use prosodic data, as well as segmental and suprasegmental characteristics like intonation, accent, pitch and manner to talk. However, each of those informations may not in itself be enough to discriminate between two different persons. The relationship made between the sources of information used to make decisions is another difference between humans and systems.

In addition to those differences, between systems and humans, some realistic problems due to system operation can be identified. The environment is a problem for the good performance of SV systems. In the acoustical signal not only the information about the speaker is found. The speech is modified in its way prior to reach the final receptor. In addition to the proper and useful information the signal contains noise from the environment and has distortions due to the communication channel which make the problem more complex. In order to attain a robust system the environmental and acquisition conditions have to be taken into account.

The problem found in the dissimilarity in the environment condition is close to the problem of lack of data in each specific condition. These problems are related to each other because of the lack of available data which can represent all the possible environments. The speech samples obtained from each speaker will never reflect all the potential utilization conditions. It is unrealistic to thinking on having samples from all possible sound environments, all signal acquisition conditions and all transmission channels.

Finally, as an extra obstacle for the SV systems we mention the well-known fact that humans' voice change with time. In general, in a SV system real application all the variations between the samples used to model a speaker and the data obtained for the test are source of errors.

To overcome all those difficulties the state of the art systems and research in SV use several techniques to compensate those differences in order to obtain more robust results. Basically, there are three such techniques. The first ones, called normalization techniques, try to make the used information independent of the employment conditions. The adaptation techniques, in other hand, try to adapt the knowledge acquired from each speaker to the new environment and utilization conditions. And finally, techniques which use some a priori knowledge to balance the lack of date in the new information.

1.1 Main Contributions

In this work we try to solve some of those problems. Two different approaches, which reflect our contribution, are presented mainly in two parts.

First, given that speech is a rich signal with several characteristics, we propose to combine in a base level several measurements, or source of information, of this signal to improve the performance of SV systems. The combination will be done at the features level because we think that it is the best way to maintain the specific relations between those source of information. We propose to use Bayesian Networks to integrate the information received from multiple measurements in a single statistical framework that keeps the conditional dependence and independence relations between all those data. To develop a SV system based on Bayesian Networks, first a study of the conditional relationships of those variables was performed. The optimal structures obtained define the first part of the SV system and the first contribution of this work.

Second, given the problems faced in a real application we propose to use adaptation techniques in our SV system. To make the system more robust to conditions of utilization we suggest to include some a priori knowledge based on a MAP approach. Therefore, we propose a technique to adapt BNs given the

the mathematical conditional independence relations of these models. Two types of relations were studied: between discrete and continuous variables. We developed two different techniques based on those type of variables.

1.2 Overview of the structure of the thesis

This work is divided in three parts. The first deals with the basics of SV systems and graphical models. First a brief description of SV systems is made. In a third Chapter the possible sources of information about a speaker are explored and tested.

The fourth Chapter deals with graphical models. Their utilization in other fields is presented. Formal definitions about graphical models are given. Directed Acyclic Graphs, BNs, receive particular attention as they are the tools used as the main probabilistic model in this work. Hidden Markov Models and their variants are studied in Chapter five as particular cases of of Dynamic Bayesian Networks.

In the second part the problems of inference, parameters and structure learning are presented. In this part the first results are presented. The obtained structures issue of a learning phase are studied and tested.

Adaptation techniques are discussed in the last Chapter. In particular, a new adaptation technique for Bayesian Networks is proposed. This adaptation is based on a distance between Conditional Probability Distributions and regression matrix used to model the conditional dependencies among all concerned variables. The document ends with some conclusions and perspectives of this work.

At the beginning of this document you can find a summary in French and at the end a small report about our participation to the NIST's evaluation as a part of the ELISA consortium.

Part I

Principles of Speaker Verification and Graphical Models

Chapter 2

Speaker Verification Systems

SV systems ([Bimbot *et al.*, 2004]) have two different phases. The first phase consists in training and the second in testing. In the training phase a model of a specific person is generated making use of some utterances. In the test phase the models generated in the training phases are used to verify the claimed identity of a speech sample. A representative Automatic Speaker Verification (ASV) system is composed of four main modules. Figure 2.1 shows the units in which the system could be divided. The first one, which it is not described because it is out of the scope of this work, is devoted to obtaining and digitizing of speech signal. This task is achieved by filtering and using an analog to digital converter (A/D).

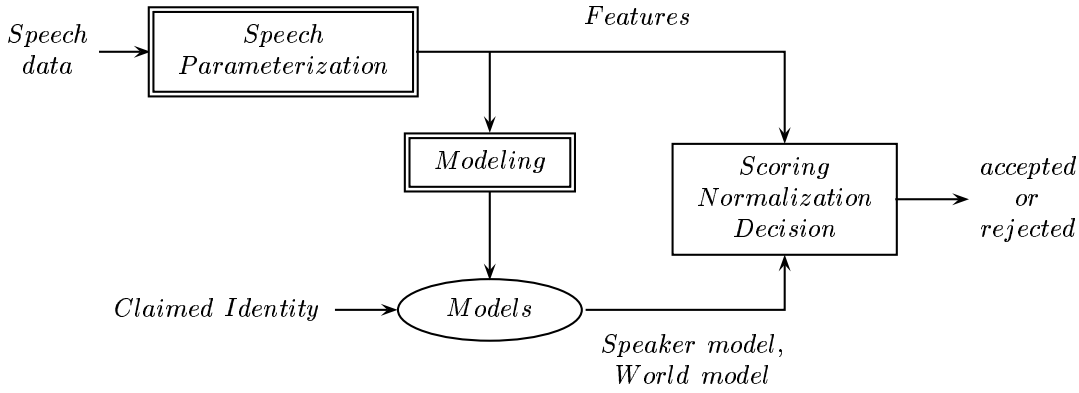


Figure 2.1: *Main modules of a Speaker Verification System. The modules in double squares are the ones where we work in this thesis.*

2.1 Speech Parameterization

The second module, speech parameterization also called Feature Extraction (this is the main topic of Chapter 3), is devoted to capturing the pertinent information from the speech signal for a specific application, in our case speaker verification. Speech is a rich and complicated signal resulting of different transformations that take place at different levels. According to the stages of oral communication four principal levels can be enumerated: semantic, linguistic, articulatory and acoustic. In each one of those levels we can obtain information about a specific speaker. For example: gender, language spoken, stress, accentuation, rythm and intonation. All of them introduce variations in the characteristics of the acoustic signal because this

signal is the result of a combination of learned habits and anatomical differences inherent in the vocal tract. Therefore, these differences can be used as discriminating information in order to differentiate between speakers. However, they need to be measured. To extract that pertinent information usually the acoustical signal is first divided into intervals (frequently between 10 and 30 ms.) called frames. Each one of those frames is then mapped to a selected multidimensional feature space. In most of the actual ASV systems the spectral information is used. Cepstral coefficients as well as the dynamic information represented in the first and second derivative (called Δ and $\Delta\Delta$ coefficients) are also used. In addition to spectral characteristics of speech some other information like pitch, which is an example of prosodic information can be employed in a SV system.

2.2 Modeling

The second unit, modeling and pattern matching, is a central module in all ASV systems. There are two different approaches for modeling: template models and stochastic models. The first one is based on template models, which were used in the beginning of the SV system and particularly for text dependent applications. The basic idea behind the template models is to measure a distance between two templates, one is the model for the frames of the speaker and the other is obtained from the frames of the uttered speech. Some examples of this kind of techniques are Dynamic Time Warping (DTW) and Vector Quantization (VQ). The second one, stochastic models, assums the observations as random vectors with a probability density function that depends on a specific speaker. Stochastic models are based principally on Hidden Markov Models (HMM), Gaussian Mixture Models (GMM) or, as is proposed in this work, on Bayesian Networks (BN) (main subject of Chapter 5) which estimate the probability density function and can be used to compute the probability of observations generated by a certain probability density function. In this case the similarity measure is the likelihood of the feature vectors obtained from the uttered speech given the speaker model. Let θ be the parameters of the model for the system and **x** the feature vector. Considering T observations of **x** the likelihood function for a GMM is computed as follows:

$$p(x|\theta) = \prod_{t=1}^{T} p(x(t)|\theta), \tag{2.1}$$

where the independence hypothesis of observations is assumed.

The choice of the actual form of the model and of the likelihood density function is affected by the conditions of use. In a system without constraints in the text of the utterance a GMM will be preferred. Otherwise, in a text-dependent system a HMM will be chosen because it allows to model the dynamic of features. Concerning BN one can say that they give more flexibility for modeling some extra relations between the features and, therefore, introduce more complexity and degrees of freedom to the models as it will be seen in part II of this work.

In general, if the system is text-independent the state-of-the-art systems use GMM as the modeling function for the distribution of characteristic vectors. The mixture of M density probabilities can then be written as follows:

$$p(x|\theta) = \sum_{i=1}^{M} w_i p(x_i; \mu_i, \Sigma_i), \tag{2.2}$$

where w_i are the weights, which verify the constraint $\sum_{i=1}^{M} w_i = 1$, μ_i is the mean and Σ_i the covariance matrix for each gaussian density that is written as follows for each component $i \in [1, M]$ of the mixture:

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{-\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}. \tag{2.3}$$

In general, the covariance matrix Σ has all elements differents of zero, that is, it is a full covariance matrix. In practice, the diagonal matrix is preferred because of computation reasons. In a diagonal matrix the inversion is just the inverse of the diagonal elements. Then inversion of a full matrix is not required.

Just to finish with this section artificial neural networks (ANN) [Prasanna *et al.*, 2004] as well as Support Vector Machines (SVM) [Wan and Campbell, 2000] should be mentioned as examples of discrimination-based learning procedures for SV.

2.2.1 Adaptation

In order to obtain consistent models, it is necessary first to use speech signals that reflect the expected circumstances of utilization, and second to use an adequate quantity of speech samples. If these two constraints are not satisfied adaptation techniques can be employed to overcome the problem. Adaptation in general and the proposed techniques for BN are the topic of Chapter 8. When not enough speech signal are available a satisfactory speaker model can be obtained by adapting a good generic model using Bayesian Adaptation Technique as Maximum A Posteriori (MAP) [Mokbel, 2001; Gauvain and Lee, 1994]. A good generic model means a model trained by employing enough data which represent the circumstances of utilization. For example, using just men speech if it is known that just men speech will be tested again a specific men model, using speech recorded in cell phones is if known that user will use cell phone.

In general, the equation employed to update the parameters of the model in adaptation techniques can be derived from the MAP approach by using some constraints in the prior distribution. This constraint is expressly chosen in such a way that the likelihood and the posterior distribution belong to the same family. Consequently the prior distribution is the conjugate of the desired posterior distribution. For example, for a gaussian distribution where the searched parameter is only the mean the searched distribution is also a gaussian distribution.

2.3 Decision

The last module in the SV system chain is the score and decision unit. As it was already presented the score is based on a distance for template models and on likelihood for stochastic model. The decision problem in a SV system can be seen as a problem of classification in two classes. The first class corresponds to the speaker S and the second class corresponds to a state called non-speaker $\overline{S}$. These classes have prior probabilities $P(S)$ and $P(\overline{S})$ and conditional probability density of the observations x given the classes $P(x|S)$ and $P(x|\overline{S})$. Using Bayes rule the a posteriori probability can be computed as follow:

$$P(S|x) = \frac{P(x|S)\ P(S)}{P(x)}, \tag{2.4}$$

where, $P(x) = P(x|S)\ P(S) + P(x|\bar{S})\ P(\bar{S})$. Figure 2.2 shows an example of impostor and client likelihood distribution. Therefore, the decision can be made on the basis of the a posteriori probabilities. Thus, the hypothesis S, the actual speaker correspond to the claimed identity is selected if $P(S|x) > P(\overline{S}|x)$. This choice is justified because a minimum error is attained with this procedure. The same equation can be written in the following form:

$$P(x|S)\ P(S)\ >\ P(x|\overline{S})\ P(\overline{S}). \tag{2.5}$$

and then:

$$\begin{aligned} \frac{P(x|S)}{P(x|\overline{S})} &> \frac{P(\overline{S})}{P(S)}, \\ \frac{P(x|S)}{P(x|\overline{S})} &> threshold. \end{aligned} \tag{2.6}$$

The decision is based on the likelihood ratio, the left side of the equation. Usually the logarithm is computed in both sides to obtain the log likelihood ratio, also known as the speaker score, which is the most common decision rule used in SV systems:

$$Score(x) = \log \frac{P(x|S)}{P(x|\overline{S})} \; > \; \log(threshold) = constant. \tag{2.7}$$

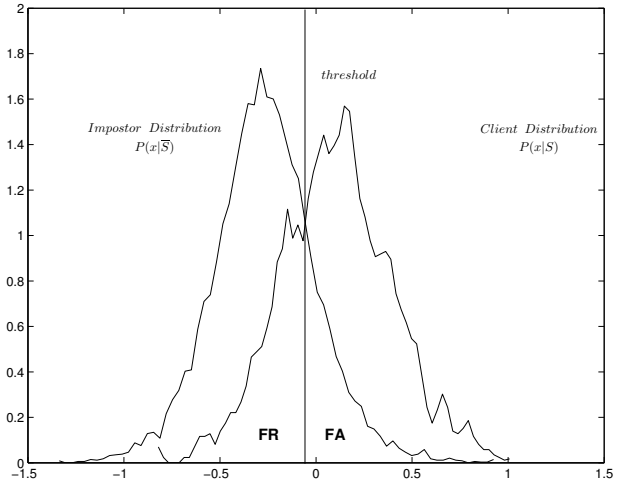


Figure 2.2: *Client and impostor probability density functions. The line in the middle represents the decision threshold. FA and FR represent the false acceptance and false reject errors areas.*

2.3.1 Universal Background Model

The non-Speaker model $\overline{S}$ in the state of the art systems is modeled by a single speaker independent background model, also called the world model. This model is trained to represent the speaker independent distribution of speech parameters. Normally, the world model is trained to represents the conditions, speech type and quality and enviroment, encountered in the test phase. For example, if in the test phase there are not cross gender tests, two world models will be trained, one for males and other for women, one model for each subpopulation. Subpopulations can also be classified by the type of hanset. Then, one wold model could be created for men using one special type of handset. All those differences should be taken into account to choose the data, which should be enough and well balanced.

2.3.2 Speaker model

The final speaker model S is in general obtained from the world model by adaptation. The basic idea of this approach is to adapt the well learned parameters of the world model, using adaptation, to finally obtaine the speaker's model. For example, if the MAP approach is used, the speaker's model is obtained using the world model as a priori. Then new prameters are obteained using the EM [Dempster *et al.*, 1997], where the E step compute the sufficient statistics of the speaker's data. Unlike the M step, the new parameters are computed by combaining the a priori, the wold model, parameters with these new sufficients statistics using a mixing coefficient (see Chapter 8).

2.4 Normalization

The decision in a SV system is made by comparing the log-likelihood ratio to a threshold as it was already in the previous section. But, this rule is based in the true probability density functions. In real application

those true functions are unknown. The computed functions depends on the real utilization conditions, the utterance and also on each speaker. Therefore the obtained speaker score has to be normalized. The first and basic technique is to normalize given the utterance length. If the score is designed by $S(x)$, where x is the utterance, and T is the number of observations in that utterance, the normalized score is:

$$\begin{aligned} \widehat{S}(x) &= \frac{1}{T}\, S(x), \\ &= \frac{1}{T}\, \log \frac{P(x|S)}{P(x|\overline{S})}. \end{aligned} \tag{2.8}$$

Score variability can come from other sources. Then, some other normalization techniques are used to adjust the score. Utilization conditions are not always the same, noise and communication channel, for example can change each time the system is used. Even, variability comes from the speaker himself. Models quality as well as the phonetic content are some of those mentioned causes.

Techniques for normalization are based on the work of [Li and Porter, 1988]. The authors showed that a variance is present on the client as well as on the impostor scores in the speaker verification test phase. Thus, the main idea of normalization is to modify the impostor scores generated by the SV system in the following form. If $S_\theta(x)$ denote the score for the model with parameters θ, the normalized score for that model is expressed in the next equation:

$$\widehat{S}_\theta(x) = \frac{S_\theta(x) - \mu_\theta}{\sigma_\theta}, \tag{2.9}$$

where σ_θ, the variance and μ_θ, the mean are the normalization parameters. Those parameters are computed as a function of the characteristic to be normalized.

2.4.1 Normalization Parameters Computed in Training Phase

Znorm [Li and Porter, 1988] normalization technique computes an impostor distribution given a model and several impostor utterances. As it can be seen this technique works in the learning phase. Then, in the test phase non extra time is needed. From the impostor utterances the parameters, μ and σ for normalizing are computed. As well as Znorm the Hnorm [Reynolds, 1996] also works in the training phase, but Hnorm gives handset-dependent distributions. This technique aims to normalize the variability that comes from a different handset utilization. Utterances employed should come from a given and unique handset. In the test phase the set of used normalization parameters is the one that corresponds to the test utterance.

2.4.2 Normalization Parameters Computed in Test Phase

Unlike above techniques, Tnorm [Auckenthaler *et al.*, 2000] uses impostor models instead of impostor utterances. At the test phase, a set of impostor models is used for scoring the input utterance in order to estimate an impostor score distribution and the set of normalizing parameters. The advantage of this technique in comparation to the Znorm and Hnorm is that the same utterance is used. Then, in this form a utterance mismatch is exclud.

2.4.3 Generating the data to Compute the Normalization Parameters

A problem in the previous techniques is the available data to generate the models and also the data to test the available models. Dnorm [Ben *et al.*, 2002] propose to generate those data from a generic client model using a Monte Carlo based method. In this case, the final score is given by the following equation:

$$\widehat{S}_\theta(x) = \frac{S_\theta(x)}{KL2(\theta, \overline{\theta})}, \tag{2.10}$$

where θ represent the client model, $\overline{\theta}$ the generic model and *KL2* the symmetric Kullback-Leiber distance between both models. The principal advantage of this technique is that it does not need extra data to normalize the scores.

2.5 Performance

As a SV system has to verify the claimed identity of a given speaker, two kinds of errors can be made as it is already been said in previous paragraphs. The first error is to accept an impostor, and is called false acceptance (*FA*), or, the second, rejects a client, called false reject (*FR*). These errors are expressed through the FA rates and FR rates respectively given by:

$$FAR = \frac{\#FAs}{\text{\# impostor acces}}, \tag{2.11}$$

$$FRR = \frac{\#FRs}{\text{\# client acces}}. \tag{2.12}$$

Those rates are used to measure the performance of a SV system. A combination of both measures called a decision cost function (DCF) can be obtained. This measure is defined as follows:

$$DCF = Cost(FR)\ P(client)\ FRR + Cost(FAR)\ P(impostor)\ FAR, \tag{2.13}$$

where the two probabilities are the prior probabilities of observe a client or an impostor, and the Cost function measure the importance given to each event. A particular case, called half total error rate (HTER) is obtained when both prior probabilities $P(client)$ and $P(impostor)$ are fixed to 0.5 and the cost functions to 1:

$$HTER = \frac{FAR + FRR}{2}. \tag{2.14}$$

Another typical parameter used to measure the performance of a SV system is the Equal Error Rate (EER) where both errors are equal.

The performances of SV systems are usually represented by Detection Error Tradeoff (DET) curves. A DET curve represents performances of detection tasks and is a standard in speaker and language recognition evaluations. In a DET curve, the two possible errors of a SV system are plotted (False Acceptance or False Alarm and False Rejection or Miss Detection) one on each axis as a function of decision threshold used. As it is seen in Figure 2.2 changes in the threshold value causes changes on the error areas FA and FR. The DET curve is generated by computing the Verification score for different threshold values. An important propriety of a DET curve, given its log normal scales, is that it should be a line if the scores of clients and impostors access follow a normal distribution. An example of a DET curve is shown in Figure 2.3.

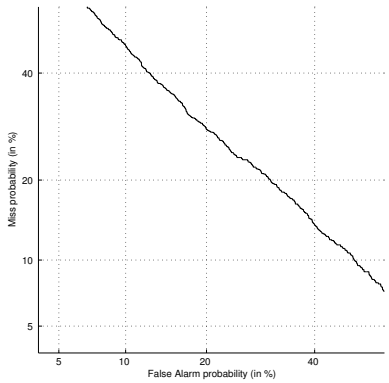


Figure 2.3: *Example of a DET curve.*

2.6 Conclusions

This Chapter was presented as a guide for this work. The basic modules (signal parameterization, modeling, decision and performance measure) forming a SV system have been briefly presented in this first chapter. Typical problems on a real SV system application were reviewed.

The next Chapter will detail the speech signal parameterization. Special emphasis on possible extra source of information obtained from the acoustical signal will be placed.

Chapter 3

Sources of Information for Speaker Verification

The speech signal carries a lot of information besides the message. Information about the speaker is present such as mood, emotive state and in particular his/her identity. Listening to somebody, it can be said if the person is a woman or a man, young or old, with a voice disorder, worried, scared, happy or not, speaking Spanish, English, German or an other language, etc. SR (Speaker Recognition) systems should use all features which capture the characteristics of the speaker in order to differentiate them from others. In this search for individual discriminant features some information could be lost. Many authors discard the prosody in speaker verification, but it is known that it carries a lot of information about the speaker identity. The suprasegmental characteristics, like intonation, accent or pitch are very important in a normal communication, specially the pitch that appears like an important factor in speaker recognition [Carey *et al.*, 1996]. However the pitch information in itself is not enough to discriminate between two different persons. Therefore knowledge comming from other sources, and not just the spectral or pitch, must be used. One example of these information, which is not often taken in account, is that which comes from the source of excitation in speech production.

In this chapter we present some possible sources of information obtained directly from the acoustical signal. We start with an overview of linear prediction (LP) analysis introducing the residual signal. Then, we present the pitch as an example of prosodic information. Then a presentation of spectral characteristics of speech is made. We finish with some results obtained using this information.

3.1 LP Analysis

The LP (Linear Prediction) Analysis [Goldberg and Riek, 2000] has become a sucsessful because of the adequacy between the proposed model and the human voice emission process in the human being. Another very important characteristic of this technique is that it has a very low computation cost. The composite spectrum effects of radiation and vocal tract is assumed to be represented by a time-varying digital filter, see Figure (3.1).

In this figure, $S(z)$ is the output signal issued from the filter $H(z)$ with the input $U(z)$. The filter function has the following form :

$$H(z) = \frac{1 + \sum_{l=1}^{q} b_l z^{-l}}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{B(z)}{A(z)}, \tag{3.1}$$

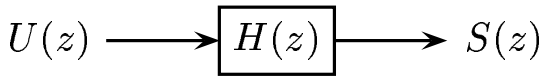


Figure 3.1: *Signal Generator Modul.*

where a_k and b_l are the digital filter coefficients. The denominator $A(z)$ is a polynomial expression in z^{-1} called the inverse filter whose first term is one and the composed elements are negatives. The parameters (p, q) define the number of poles and zeros of the filter. It could be imposed $q = 0$ to obtain a filter with only poles. Then, a designed output $S(z)$ imposes some changes in the input signal $U(z)$ and the a_k coefficients. Therefore the equation of this filter can be written as follows:

$$H(z) = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{1}{A(z)}. \tag{3.2}$$

Finally, the input signal could be computed from the wished output signal in the following form :

$$U(z) = H^{-1}(z)S(z) = A(z)S(z). \tag{3.3}$$

From equation (3.3) and the inverse transform Z^{-1} the equation in the time domain is written as :

$$u(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \; \forall n, \tag{3.4}$$

This equation highlights the prediction property that permits to obtain the values of $s(n)$ from a linear combination of the p last samples given the minimization criterion of the input signal $u(n)$ energy. Moreover it can be seen that the filter is an unstable filter since it has an output even if at the entrance the signal is null. The LP analysis is performed to determine the predictor coefficients a_k, which represents the vocal tract model of the speaker, directly from the speech signal.

Computing (Developing) the energy (E) shows in the next expression :

$$\begin{aligned} E &= \sum_{n=-\infty}^{\infty} u^2(n) = \sum_{n=-\infty}^{\infty} \Big(s(n) - \sum_{k=1}^{p} a_k \; s(n-k) \Big)^2, \\ &= \sum_{n=-\infty}^{\infty} \Big(s^2(n) + \Big(\sum_{k=1}^{p} a_k \; s(n-k) \Big)^2 - 2 \; s(n) \; \sum_{k=1}^{p} a_k \; s(n-k) \Big) \end{aligned} \tag{3.5}$$

and minimizing this energy given the p filter coefficients a_i the searched solution is obtained as follows :

$$\sum_{k=1}^{p} a_k \phi(k-i) = \phi(i) \; \forall i \in [1, p], \tag{3.6}$$

where $\phi(i)$ is the autocorrelation function of a real signal with finite energy :

$$\phi(i) = \sum_{n=-\infty}^{\infty} s(n)s(n-i) \; \forall i. \tag{3.7}$$

It is not possible to use the autocorrelation function $\phi(i)$ in practice because of the limited number of available samples at a given time. This problem can be solved in two different ways. In both options it is just considered a small part of $s(n)$, with a window of finite length. The first choice is called the autocorrelation technique and admits the distortion effects caused by the convolution between the window and the signal. The second option, called the covariance technique, uses the signal energy E instead of only the signal $s(n)$ to reduce the mentioned distortion effect.

The input signal $x(n)$ used in the autocorrelation techniques is the weighted version of the original one $s(n)$ using the window weights $w(n)$:

$$\begin{cases} x(n) = w(n)s(n) & \forall n \in [0, N[\\ x(n) = 0 & \forall n \notin [0, N[\end{cases}$$

Using this new signal $x(n)$ the autocorrelation (3.7) function becomes :

$$R(i) = \sum_{n=i}^{N-1} x(n)x(n-i) = \sum_{n=i}^{N-1} w(n)s(n)w(n-i)s(n-i), \quad \forall i \in [1, p], \tag{3.8}$$

3.2 Information from the Source of Excitation

Human speech is produced by vocal organs like the vocal cords, the larynx etc. The most important fact here is that all those organs are specific to the speaker. When someone speaks, the air flow is forced through the glottis between the vocal cords and the larynx to the three main cavities of the vocal tract. Therefore the excitation signal produced has the speaker's mark and can be an extra information about the speaker. The Linear Prediction (LP) analysis provides a method for separating the vocal tract information from excitation.

3.2.1 Residual

The LP residual [Thévenaz, 1993; Faúndez-Zanuy and Rodríguez-Porcheron, 1998] is spectrally flat, because the vocal tract shape is removed by creating an inverse filter $A(z)$. This filter is obtained using the LP coefficients obtained, in our case, by solving the Yule - Walker equation. Actually the predicted residual is the excitation signal normalized by the prediction gain G as reflected in the next equation that is derived from equation (3.4) :

$$\hat{u}(n) = \frac{1}{G}\Big(s(n) - \sum_{k=1}^{p} a_k s(n-k)\Big) \ \forall n, \tag{3.9}$$

where G is given by the expression :

$$G = \sqrt{\phi(0) - \sum_{k=1}^{p} a_k \phi(k)}. \tag{3.10}$$

As an example Figure (3.2) shows a typical voiced signal with a good periodic structure at the top and its residual signal computed using the equation (3.9) at the bottom. It can be seen that the periodic structure of speech is represented by periodic pulses in the LP residual of voiced speech.

3.3 Prosodic Information

As it was already said prosodic information is another source of information. Prosody has to do with speech features whose domain is not a single phonetic segment, but larger units of more than one segment, possibly whole sentences or even longer utterances. This is the classical definition of prosody and from it, it

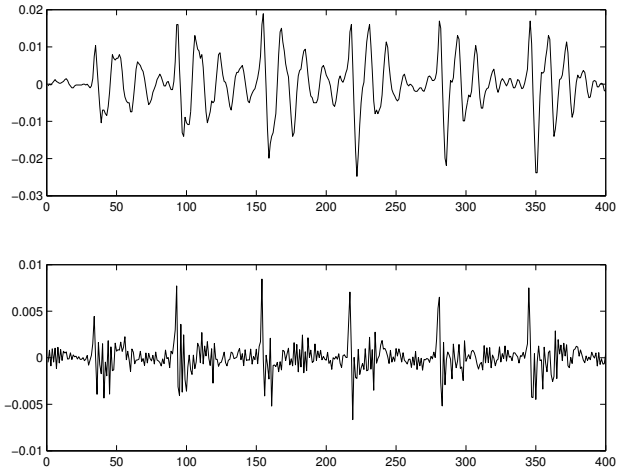


Figure 3.2: *Example of a voiced signal (top) and its residual signal (bottom).*

can be said that prosodic phenomena are supra-segmental. They appear to be like features which structure the speech flow. They are perceived as stress, accentuation, rythm and intonation. These characteristics are well adapted for speech segmentation and for SV (Speaker Verification) given that they are features that characterize the speaker all along the speech.

In order to make a classification of prosodic features it is necessary to take into account four manifestation levels according to the stages of oral communication. From the intention level, linguistic and paralinguistic, the prosody is seen as an element that could help coding certain intentions expressed in the utterance. Linguistic expressions use only language signs but prosody means can help on communicate linguistic distinctions, relating different linguistic elements or defining transitions between words. On the other hand, paralinguistic expressions use non-verbal vocalizations like interjection-like expressions as well as expressions that make the utterance sounds urgent, worried, etc. In the articulation level, prosody features are modifications of articulatory movements which are observed only with sophisticated machinery. The most important level in this work is the acoustic level. The acoustic realizations of prosody can be observed and quantified using acoustic signal analysis. The main acoustic parameters are the fundamental frequency, intensity and duration. The last level is the perceptual level. Here, the prosodic information carried by the acoustic signal is decoded by the ear and brain obtaining linguistic and paralinguistic informations.

Hereafter, a study of the fundamental frequency (f_0) is given because it is one of the speech characteristics used to model the speaker in our work.

3.3.1 Fundamental Frequency

The fundamental frequency, or pitch, of a periodic sound is that sinusoidal component of the sound which has the same period as the periodic sound [Moore, 1982]. But in this work the fundamental frequency is not the perceived, subjective tonal quality of a complex sound. The pitch is a property of voiced speech which has a closed relation with the glottis movements. The organ opens and closes in a particular fashion, giving (imparting) a periodic character to the excitation. The pitch period, called T_0 is the time span between two openings of the glottis, and the fundamental frequency is the reciprocal of the pitch period, $1/T_0 = f_0$.

The physical limitations of the human vocal cords restrict the pitch range to frequencies between 50 and 300 Hz. Men generally have lower pitch frequencies and women and children have the higher part of that

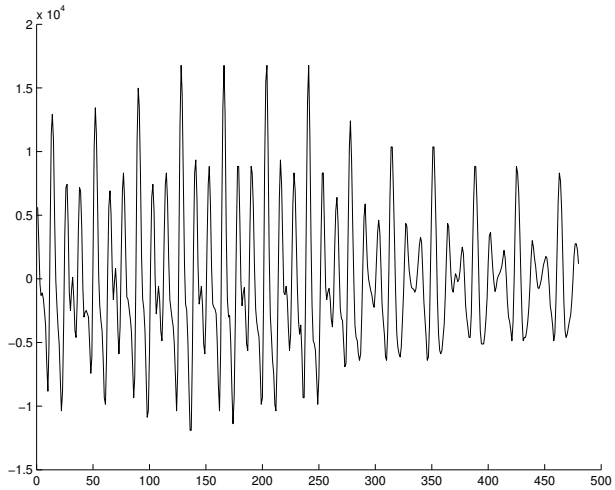


Figure 3.3: *Example of a voiced speech signal.*

range. As already said, pitch is an important prosodic characteristic that carries paralinguistic informations by the rise and the fall of the frequency.

Estimation of pitch entails a big difficulty on account of the quasi-periodic nature of voiced excitation. Not only detection of the slowly variation of the excitation waveform is a problem, also the time point chosen for period measurement could change the measure. The vibration of the vocal chords can even be quite non periodic and harmonics or sub harmonics of the fundamental frequency can appear more prominent than those of the actual pitch frequency.

3.3.2 Autocorrelation Pitch Estimation

In this section will be discussed one of the most popular and fastest approach for pitch computation [Goldberg and Riek, 2000]. This technique tries to locate the periodicity in the time domain using the correlation function which measures the degree of similarity between two signals. Actually, the autocorrelation function measures the similarity between the signal and its shifted version. The maxima of the autocorrelation function takes place at the moments that the shift coincide with the pitch period of the original signal.

As already mentioned (section 3.2), in real conditions it can just be computed the short time autocorrelation function :

$$R(i) = \sum_{n=1}^{N-1} s(n)s(n-i), \tag{3.11}$$

Figure(3.3) depicts a segment of 60 ms of a voiced speech signal sampled at 8 kHz, and Figure (3.4) displays the autocorrelation of the same segment.

From the autocorrelation function it can be seen that the second maximum is about the sample number 38. Then the pitch period is also about 40 samples. This lag of 38 samples corresponds to a pitch period of $T_0 = 4.75$ ms and a pitch frequency of 210 Hz. A local maximum also appears at a lag of 76 samples. This value shows the good match when the shift is twice the pitch period. The maximum value for the autocorrelation function is always at lag 0 since the match between the signal and the non shifted signal,

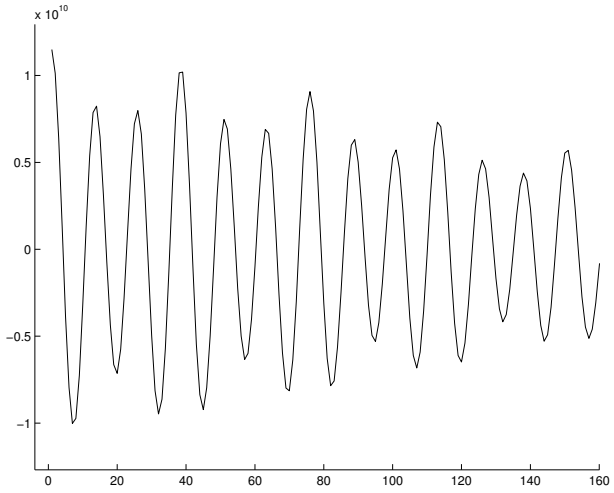


Figure 3.4: *Autocorrelation function of a voiced speech segment.*

itself, is maximum. This value $R(i)$ is the computed energy when $i = 0$ in the equation (3.11).

Since speech is not a pure periodic signal, and vocal tract resonances produce some other maxima, pitch computation directly from the autocorrelation signal can results in multiple local maxima. The suppression of local maxima could be made using the method which consists on center clipping the signal before computing the autocorrelation function. Amplitude values of the original signal under a fixed positive value C_L and above the negative value $-C_L$ will be zero for the center clipped signal. The rest of the signal is equal to the original minus or plus the fixed value C_L. Figure (3.5) shows a diagram of a center-clipped speech signal obtained from the signal in the figure (3.3). The autocorrelation function of the clipped waveform is in Figure (3.6). A signal plus its pitch frequency is depicted in Figure 3.7.

3.4 Spectral Information

In general the most employed source of information in speech processing is the cepstrum which is derived from the spectrum of the signal. The cepstrum is defined as the inverse discrete Fourier transform of the log of the magnitude of the discrete Fourier transform of the input signal $s(n)$. The discrete Fourier transform (DFT) is the most used transform in the speech domain. The DFT is a Fourier representation of a sequence of samples of limited length. The DFT and the inverse (IDFT) are defined as :

$$\begin{cases} S(k) = \sum_{n=0}^{N-1} s(n) e^{-j\frac{2\pi}{N}kn}, \\ s(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k) e^{j\frac{2\pi}{N}kn}, \end{cases}$$

and the cepstrum is expressed in the next equation that uses the Fast Fourier Transform as a fast algorithm to compute the DFT :

$$Cepstrum(d) = IFFT(\log_{10} |FFT(s(n))|), \tag{3.12}$$

The index d is called the quefrency of the cepstrum signal. A value in d corresponds to a periodic component in the signal with frequency $1/d$.

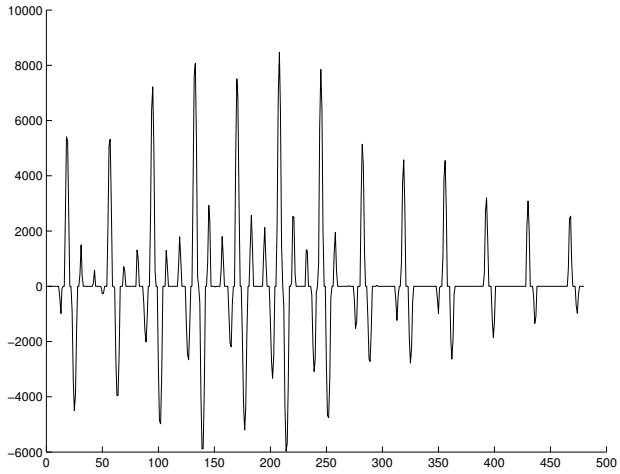


Figure 3.5: *Center clipped signal.*

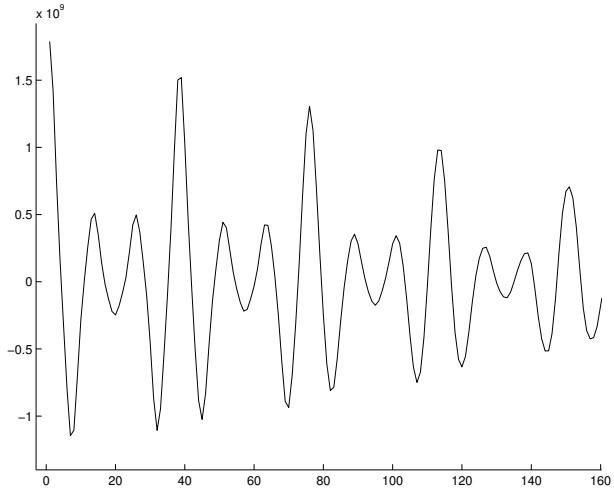


Figure 3.6: *Auto-correlation function of the center clipped signal.*

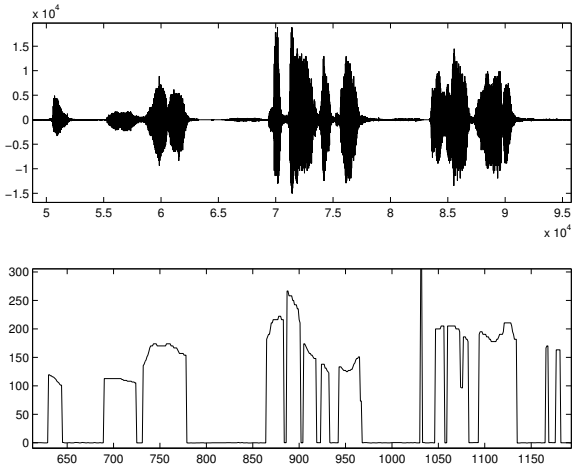


Figure 3.7: *Signal plus pitch.*

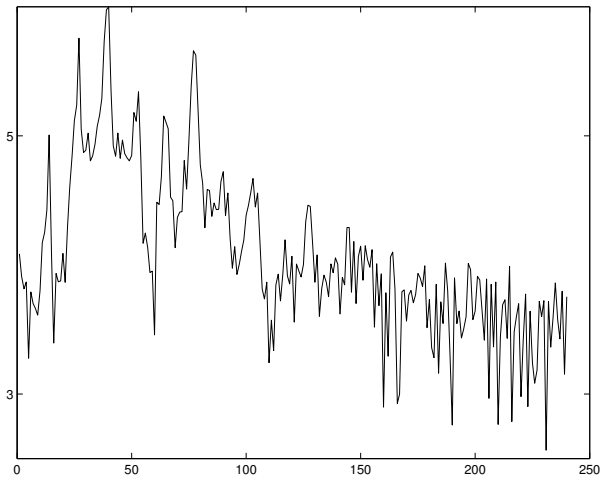


Figure 3.8: *Example of spectrum.*

Figure (3.8) depicts the log magnitude spectrum and Figure (3.9) displays the corresponding cepstrum for the speech signal in Figure (3.3).

3.4.1 Properties of the cepstrum

An important property is the linear effects of the obtained cepstrum values when the signal is passed through a linear time invariant filter. The cepstrum of the filtered signal is equal to the sum of the cepstrum of the original signal and the cepstrum of the linear filter. Then, if the linear filter is considered time invariant and the longterm spectrum of the speech is considered to be flat, the cepstrum from the input signal can be obtained by the subtraction of the time average from the output cepstrum. For this operation it is assumed that the input signal is long enough such that the input signal energy is distributed over the entire range of the spectrum. This property is used to eliminate the communication channel (the linear filter) effects in the original signal. But the most important property of cepstral analysis is the uncorrelated relation between

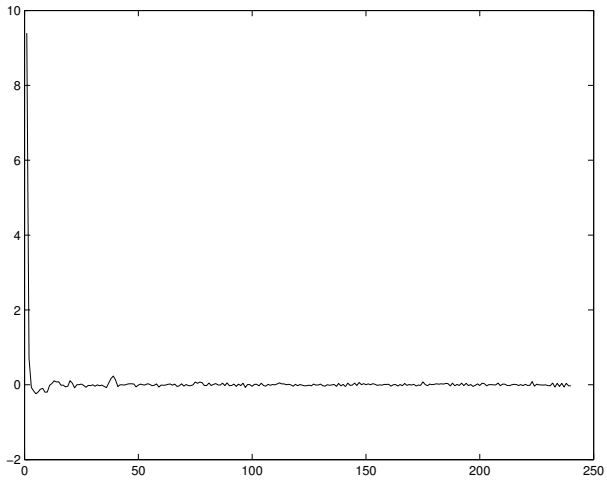


Figure 3.9: *Example of cepstrum.*

the cepstral coefficients.

3.4.2 Mel Cepstrum

Theory of human audition is essential in the spectral analysis. It is known that human inner ear, basilar membrane, works like a spectrum analyzer with a resolution that is characterized by critical bands that does not follows a linear scale. Those critical bands define a subjective criterion of the content in frequency of a given signal, that is, a bandwidth at which that subjective response is significantly different.A measure of those critical bands establish two different scales: Bark and Mel. Both, Bark and Mel are units based on the perceptual frequency that increase logarithmically with frequency as is shown in the next equation for the Mel scale:

$$M = \frac{1000}{\log 2} \log\left(1 + \frac{f}{1000}\right),$$

where, M is in Mel units and the frequency f in Hertz.

This nonlinear frequency perception scale has given place to a model that takes into account the subjective frequency perception in humans. One approach to attain this scale is using a filter bank. The filters are spaced in the nonlinear, for example the Mel scale.

If the power coefficients of the spectrum at the output of the bank filter is $\hat{S}(k)$ for $k = 1, 2, \ldots, K$ and K such filters, the Mel-frequency cepstrum $\hat{c}(n)$ is:

$$\hat{c}(n) = \sum_{k=1}^{K} (\log \hat{S}(k)) \cos\left[n\left(k - \frac{1}{2}\right) \frac{\pi}{K}\right], \quad n = 1, 2, \ldots, L,$$

where L is the number of cesptrum coefficients.

3.4.3 Temporal Spectral Information

The cepstral information can be improved introducing temporal information of cepstrum. This extra information is obtained in the first and second derivatives of cepstral coefficients. To introduce the temporal variable in the cepstral analysis a t coefficients is added to the notation. Then, a cepstral coefficient $c(n)$

at time t will be $c^t(n)$. In general the t factor refers to the frame instead of the time instance. The time cepstral derivative is approximated as follows:

$$\frac{\partial}{\partial t}\left[\log\ |S(\exp^{jw}, t)|\right] = \sum_{m=-\infty}^{\infty} \frac{\partial c^t(m)}{\partial t}\ \exp^{-jwm}$$

Given that the cepstral coefficients are a discrete time representation using just a first or second order difference results in a noisy derivative. Then, a better option is to approximated the derivative inside the sum in 3.13 by a orthogonal polynomial fit over a finite length window as is showed in the following equation:

$$\frac{\partial c^t(m)}{\partial t} = \Delta c^t(m) \approx \mu \sum_{k=-P}^{P} k c^{t+k}(m),$$

where μ is a normalization constant and $(2P+1)$ is the number of frames used to compute the derivative.

3.5 Contribution of Source of Information in Speaker Verification

Throughout this section the proposed source of information will be tested. State-of-the-art SV systems are based on GMM models using MAP adaptation (see section 2.2.1 and chapter8). Then we will start modeling the different sources of information with GMM.

First, the database will be described as well as the specific parameters obtained from the speech signal. The SV system evaluation have been done in the context of NIST 2004 speaker recognition evaluation [NIST's 2004 Speaker Recognition Evaluation, 2004](for more details about our work in relation to the NIST's evaluation see Appendix A). From the 28 different combinations of training/test conditions we had worked on the core test. The core test consists of 1 side conversation for training and also 1 side conversation to test. Each conversation side consists of the last five minutes of a six minute conversation.

3.6 Database

The data, Mixer Corpus, is taken from the conversational speech data collected in the Mixer Project using the Linguistic Data Consortium [Switchboard Corpora LDC, 2004; Campdell *et al.*, 2004] new "Fishboard" platform. This database is a designed speech corpus collection for speaker recognition evaluations in a text-independent and channel-independent conditions. The data is mostly conversational telephone speech in English but there are also some speech in languages other than English, like Arabic, Russian, Spanish and Mandarin, normally spoken by bilingual subjects. When the non-English language is pared the speaker use it to communicate. Each speaker is asked to send the transmission and handset type for each call as well as the subject personal identification number. A topic, one for day, is proposed to the speakers, who did not know each other, but the speakers can change the topic during the conversation. Each conversation is not echo canceled and the silence intervals were not excised in the original database. The data have an automatic speech recognition (ASR) transcription with a Word error rate (WER) of about 20-30%. for the other languages the English recognizer was run on. The database is divided into training data (about 246 male target speakers and 370 female target speakers), and test data (about 11507 test segments for male and 14717 for female). All speakers participating in up to 25 calls of at least 6 minutes duration. All presented test have been done in the male database only.

Since there are not enough training data for each speaker, adaptation methods are applied to compute every Target Speaker Model. For this purpose, the system starts from an universal model (UBM) which is then adapted to the client speaker. UBM models have been created using part of the 1999, 2000 and 2001 cellular evaluation datasets. This database is similar to the database already described, but each conversation is echo canceled in the original database and there are just English conversations. The training data for

a speaker consist in about two minutes of speech, excerpted from a single conversation. Actual duration is, however, constrained to lie within the range of 110 to 130 seconds. Each test segment is extracted from a one minute excerpt of a single conversation and is the concatenation of all speech from the subject speaker during the excerpt. The duration of the test segment therefore varies, depending on how much the speaker spoke. Therefore, the effective speech duration lies between 15 and 45 seconds.

3.7 Speech Parameters

On account of their relevance, their relative easy computation and independence, the **cepstral**, the **residual signal**, the **pitch** and the **energy** of the signal have been chosen as the main variables to build the speakers models.

The training and test parameter vectors consist of a set of four types of parameters present all the 10 ms over a 20 ms window. The first vector is a 24-dimensional Linear Prediction Cepstral Coefficients obtained as follow: 12-dimensional LPCC plus adding their first derivatives (Δ) yielding the $SLPCC$ (Signal Linear Prediction Cepstral Coefficients) with a channel normalization with Cepstral Mean Subtraction and Reduction or using Feature Warping ([Pelecanos and Sridharan, 2001]). Similarly, the LP residual signal is represented using Mel Frequency Cepstral Coefficients in a 24-dimensional vector yielding the $RMFCC$ or using LPCC yielding the $RLPCC$, both with Cepstral Mean Subtraction and Reduction. And finally the frame pitch F_0 and the frame energy E.

Performance of the systems are shown using DET (Detection Error Tradeoff) curves. The decision score is directly based on the log-likelihood ratio between the target speaker and the UBM over all the frames without any kind of normalization.

3.8 Source of Information with GMM

The first set of experiments in this chapter examines the performance of a GMM system adapted using Maximum A Posteriori (MAP) techniques with the four variables described above, $SLPCC$, $RMFCC$, F_0 and E as input. The goal of these experiments is to investigate the recognition performance as a function of different number of components in the mixture and of different information variables used as an input for the system.

In this section four experiments were devised :
For the first case (**Experiment I**) a single GMM was trained and tested with the $RMFCC$ coefficients without the Δ part. In the second case (**Experiment II**) the GMM was trained and tested with all the $RMFCC$ coefficients (including Δ). A third part (**Experiment III**) uses the F_0 coefficients. The fourth part (**Experiment IV**) uses the energy E. The fifth (**Experiment V**) uses the $SLPCC$ coefficients including the Δ coefficients and the last experiment (**Experiment VI**) uses the four variables concatenated in a single vector.

3.8.1 Experiment I

For the first experience, the system uses the $RMFCC$ coefficients **without** the Δ coefficients, the results are shown in Figure 3.10. The experiments were done with 8, 16, 32, 64 and 128 gaussian components in the mixture. The figure shows the performance of the system as a function of the number of components in the mixture. Table 3.1 gives the obtained EER scores.

The results shown in figure 3.10 are unusual because normally increasing the number of components always decrease the EER score. To explain those results we should remember the structure of the residual signal 3.2 and also think about the quality of the speech signal. The quality of the speech in a normal

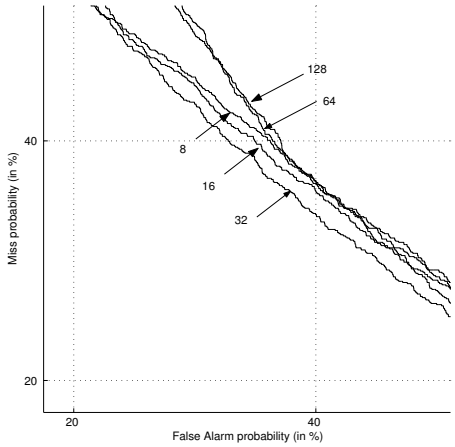


Figure 3.10: *Experiment I, DET curves for GMM system using $RMFCC$ coefficients without Δ.*

Table 3.1: Experiment I, EER scores for the GMM with $RMFCC$ without Δ as a function of number of gaussians.

	8	16	32	64	128
score	37.75	37.23	36.11	37.91	37.97

cellphone call is not really good. Then, the residual signal is almost noise and just a important periodic form is stand out. Increasing the number of components to model this signal could just increase the gaussians used to model the presented noise in the residual.

3.8.2 Experiment II

For the second experiment the Δ coefficients from the $RMFCC$ coefficients were used. Figure 3.11 and table 3.2 depict the results.

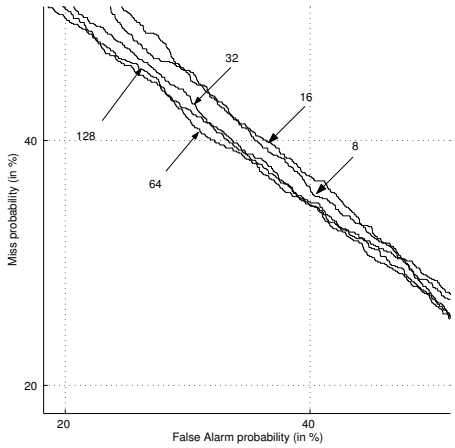


Figure 3.11: *Experiment II, DET curves for GMMs with $RMFCC$.*

From the obtained DET curves in **Experiment I** and this one **Experiment II**, it can be seen the good effect brought by the Δ coefficients as well as the gain in the false alarm detection. However this performance gain goes in a direct relationship together with the increase in the computation time and input vector

Table 3.2: Experiment II, EER scores for the GMM with $RMFCC$.

	8	16	32	64	128
score	37.89	38.27	36.76	36.44	36.96

size. Once again, the structure of the residual signal can help to explain the results. Then the extra gaussian components are best employed modeling the dynamics carry in the Δ coefficients. This effects is also seen in the gain obtained in false alarm detection score.

3.8.3 Experiment III

In this part the pitch F_0 contribution in itself is studied. Results are shown in Figure 3.12 and Table 3.3.

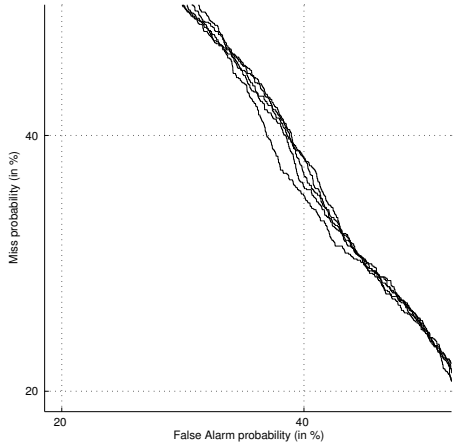


Figure 3.12: *Experiment II, DET curves for GMMs with F_0.*

Table 3.3: Experiment II, EER scores for the GMM with F_0.

	8	16	32	64	128
score	37.64	38.71	39.02	39.02	39.12

Those results show that pitch is an important source of information. Also, it should be remarked that not only the voiced parts of speech were used. The unvoiced parts of the speech signal were modeled using a random noise with a Gaussian distribution. Then, in this way the unvoiced parts will represent an state which will be represented by a Gaussian.

3.8.4 Experiment IV

In this part the energy E contribution in itself is studied. Results are shown in Figure 3.13 and Table 3.4.

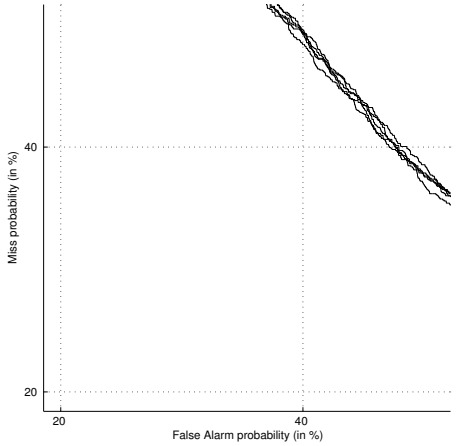


Figure 3.13: *Experiment II, DET curves for GMMs with E.*

Table 3.4: Experiment II, EER scores for the GMM with E.

	8	16	32	64	128
score	44.55	44.55	44.93	44.92	45.07

These results only show that energy by itself has no enough information to differentiate between two speakers but still there is some information. Then, energy could be combined with other sources of information.

3.8.5 Experiment V

In the third experiment of this first part the vector $SLPCC$ was employed as input. Again, a GMM system was used and tested with different number of components (8, 16, 32, 64 and 128). Figure 3.14 and table 3.5 show the results.

Table 3.5: Experiment III, EER scores for the GMM with $SLPCC$.

	8	16	32	64	128
score	31.42	29.34	27.95	27.81	27.11

These results show the importance of spectral information. Using only 8 components in the GMM the score is better than the score obtained in any result presented until now. This result is also well explained taken into account the quality of the signal.

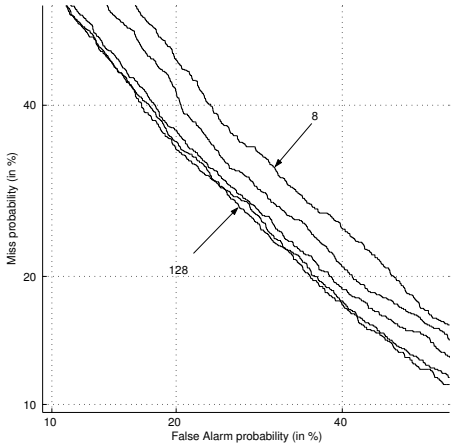


Figure 3.14: *Experiment III, DET curves for GMMs with $SLPCC$.*

3.8.6 Experiment VI

A last experiment was performed using a single input vector composed of all variables, $SLPCC$, $RMFCC$, F_0 and E, that is, all the information is combined in a single vector. It should be remembered, that pitch component in the final vector for unvoiced parts is represented by a random noise with zero mean. This time the mixtures were composed of 4, 8, 16, 32, 64 and 128 gaussians. Figure 3.15 and table 3.6 show the results.

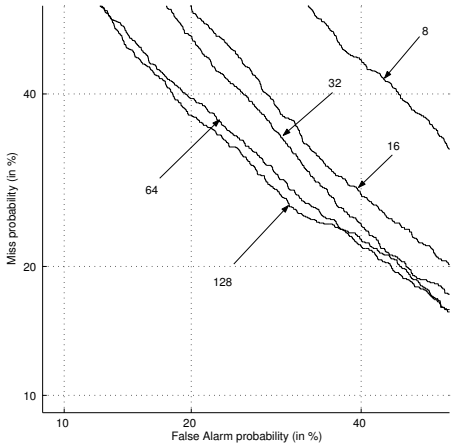


Figure 3.15: *Experiment IV, DET curves for GMMs with all variables.*

Table 3.6: Experiment IV, EER scores for the GMM with all variables $\{SLPCC, RMFCC, F_0$ and $E\}$.

	8	16	32	64	128
score	42.59	33.30	31.82	29.47	28.52

This vector combines all the variables in a base level but it does also mix the information. All the variables are in the same vector space and each characteristic is not well modeled. From the last results, all shown in table 3.6, a conclusion is that to build a single vector with all the information is not the best option in order

to increase the performance. Therefore another way for combining those variables should be investigated.

Table 3.7: Experiment I, II, III and IV, EER scores for the GMMs.

	8	16	32	64	128
Exp I RMFCC - Δ	37.75	37.23	36.11	37.91	37.97
Exp II RMFCC	37.89	38.27	36.76	36.44	36.96
Exp III Pitch	37.64	38.71	39.02	39.02	39.12
Exp III Energy	44.55	44.55	44.93	44.92	45.07
Exp III SLPCC	31.42	29.34	27.95	27.81	27.11
Exp IV All	42.59	33.30	31.82	29.47	28.52

3.9 Conclusions

In this chapter the spectral (cepstral), prosodic and residual information obtained from an acoustical speech signal were presented. At present it is know how to select those variables but it is not known how to integrate of all of them in a well suitable model. By presenting those possible source of information, this chapter laid a reason to try to combine them in such a form that a SV system becomes more robust. This possible combination also gives a foundation of subsequent analyses.

Obtained results can not be easily compared to results at the state of the art. Our best results presented using the cepstral information ($SLPCC$) and a GMM for modeling are far of the state of the art of about 15%. This difference comes from the front-end process, the number of Gaussian components used (state of the art systems use 2048 gaussian components), the quantity of speech signal used to build the world model (the state of the art systems uses about 15-20 hours of speech and we use only four) and also from the normalization technique employed (we do not use any kind of normalization). In fact, we are interested meanly into the potential of BN like an statistical tool to model speech signals.

In the upcoming chapters graphical models are described. They offer the possibility to combine different variables preserving their conditional dependencies. We start with an introduction to Graphical models is given.

Chapter 4

Graphical Models Concepts

4.1 Introduction to Graphical Models

One of the charms of studying Bayesian networks consists in the curious range of things to which they apply! [Gilles, 2002]

4.1.1 Origin of Graphical Models

Originally graphical models where developed in several areas of science. For example, in statistical physics its origin can be found in the work of Gibbs [Gibbs, 1902]. In this area the objective was to study a large system of mutually interacting particles, for example the atoms of a gas or solid. Total energy of the system is composed by an external potential plus a potential due to interactions between groups of particles. These interactions depends on the position and state of each particle. Usually it is assumed that just particles at sites close to each other, the neighbors, interact. An undirected graph was used to model the relationship between the neighbors in the system using the Gibbsian distribution

$$p(x) = \frac{1}{Z_T} e^{\frac{-E(x)}{T}}, \tag{4.1}$$

where T is the system temperature, $E(x)$ is the total energy when the system is in the state x, and Z_T is a normalizing constant.

Another origin of graphical models can be found in genetics. Wright in [Wright, 1921] used it for studying heritable properties of natural species in his so-called path analysis. Graphical models were used to model direct relationships with arrows moving from parent to child. These ideas of path analysis were later taken in economics and social sciences [Wold, 1960; Blalock, 1971].

The Work of Bartlett [Bartlett, 1935], the notion of interaction in a three-way contingency table, is the third origin of graphical models. At first sight, this work is not related to graphical models but the notion of interaction is identical in a formal way to the same notion used in statistical physics.

4.1.2 Bayesian Networks in others fields

Bayesian Networks (BN) are used in many scientific fields, in many ways and for many things because of its attractive formalism for representing uncertain knowledge. Examples include the use of BN in biology, engineering, sound processing, bioinformatics, speech processing, etc.

Medical, Biology, Management, Filtering

A curious example can be found in [Jensen, 1996] (pp. 36-37). It comes from the field of veterinary science. Here, one wants to know if a cow is pregnant. A BN is used to model the relationships between the variables : blood test, urine test and scanning. Another example in the biological field is described in [Stetter *et al.*, 2003]. In this paper, the authors try to understand the regulatory genetic network, the interactions between proteins and gnome, by means of BN. By learning the network structure they attempt to observe the relationship between genes groups. Those relationships can be the underlying cause of a specific global gene-expression pattern. For example, an observed gene expression pattern could be the evidence of a fault in a global control function. A very interesting work is presented in [Robles *et al.*, 2004]. BNs are used to combine classifiers for protein research. At a first level, a set of specialized classifiers called the component classifiers, give each one a prediction. At a second level the BN combines all those predictions to give a final predicted class. In [Acid *et al.*, 2004] an emergency medical services is modeled with a BN. The management of health services is the goal of this paper. BNs serve to model the relationship between some variables like : financing, date of admission, cause of admission, pathology and date of discharge. Four different algorithms were used to learn the independence relations among the variables. Then, those structures were tested with some records issues of the same data base. The obtained results show that the best structures recovered by the learning algorithms do not obtain the best results in the test part. The data stored in local database, also known as Web databases are the objective of [Calado *et al.*, 2004]. BN are used to model the structured queries derived from an initial user input. The events : queries, keywords and documents, which are not independent, are the variables in the BN. Given that documents and queries are composed of keywords the problem of document retrieval can be seen as a probability computation of documents given the query. [Andersen *et al.*, 2004] shows how to use a DBN for filtering. In this paper the problem of fault detection in a water-tank system is studied. A DBN is built assuming a Markovian stationary system and using Kalman Filtering. The variables present in the structure are for example the indications of measurement failures, pressure, flow and pipe resistance.

Speech Processing

BNs have also been used in speech processing and related fields. In [Fernandez and Picard, 2003] an example of classification of driver's speech under stress is given. This work tries to model speech in the context of stress for improving the robustness of speech recognizers. In the field of sound processing the paper [Kashino and Murase, 1999] uses BN for auditory scene analysis, which means recognizing many acoustic events occurring simultaneously. In this paper the sound source identification with the music stream information is defined as the estimation of the posterior probabilities of sound sources when each note is observed. In [Dielmann and Renals, 2004] Dynamic BNs (DBNs) are used to segment meetings into a sequence of meeting actions. The audio actions utilized in this paper are five : monologue, dialogue, note taking, presentation and presentation at the white-board. Those actions are the hidden variables in the model. The inputs or observed variables are obtained from a microphone array and from an single microphone for each speaker. The baseline system is an Hidden Markov Model (HMM), where hidden variables represent the meeting actions and the observed variables are the merged vector obtained from the microphone array and the single microphones. DBNs are used to represent each actions as an individual hidden variable with only one or two observed vectors (one from the microphone array and other from the single microphones).

Audio Visual Speech Recognition (SR) systems are also built using BN. In [Nefian *et al.*, 2002] both information flows are modeled by a coupled HMM, one HMM for each modality. The relationships between both modalities are given by edges which joint the hidden states in both HMMs. In [Gowdy *et al.*, 2004] a more complex model based in a SR system [Bilmes *et al.*, 2001] is used. That model is extended to incorporate the visual flow of information in such a way that the final model takes into account the asynchrony between the visual and audio modalities. In this model the signal to noise ratio (SNR) is taken into account given more or less importance to the audio flow using an exponent. A very close paper [Hershey *et al.*, 2004] talks about Graphical Models for Speech detection and enhancement. This paper is close to the pre-

viously refereed paper since the authors propose the utilization of both audio and visual informations in a single model. The proposed system fuses audio and video by learning the dependencies between the speech signal and the location of lips during speech.

Speech Recognition

One can speak about two different approaches in SR. On the one hand the explicit models represent all the underlying variables and control mechanisms. That means, there are variables for representing words, phonemes, occurrence of phonemes, transitions, etc. The interactions between those variables are expressed, for example, in the structure of a graphical model. On the other hand, implicit models, use only a single variable to embed all the control variables and variables which represent words, phones etc.

A classical example of an implicit model is an HMM. The hidden variable in this model represent all the lack of knowledge about the underlying variables in the speech production process. A little more complex model is an auto-regressive model. In [Wellekens, 1987] the author talks about a model where the observations are joined by an arc by defining a new emission probability which takes the correlation between successive features vector. In almost the same way the paper [Kenny *et al.*, 1990] proposes to model the correlation between successive frames. In [Bilmes, 1999; 2003] Buried Markov Models (BMM) are presented. A cross-observation dependencies are added between observation elements to increase both accuracy and discrimination. HMM2 [Weber *et al.*, 2002] models observations in a single time slice with a fixed length HMM. Each hidden state in the main chain "generates" a secondary HMM. For example, if the observations contains spectral information the secondary HMM works in the frequency dimension of speech. Another model which tries to represent the dynamics in the frequency domain is treated in [Daoudi *et al.*, 2003; 2000]. DBN are used to model the dependencies between the speech bands, allowing "communication" between bands. Even if in this last work hidden variables models represent frequency information they still keep too much information about the speech production process in a few variables.

The model described in [Zweig, 1998] is an example of explicit model. In this work the author proposes to use hidden context variables which represent articulatory variables. In [Stephenson *et al.*, 2000] the state of the articulators are included using some extra variables. A special characteristics in this system is that the articulators variables are observed in the training step and hidden in the test step. Another approach is to use the pitch as an auxiliary information [Stephenson *et al.*, 2001] with similar conditional independencies between observations and extra variables. Gender is a supplementary variable used with BN [Markov and Nakamura, 2003]. In this paper a hybrid system is proposed. First a BN is used to add the gender and pitch variables to the spectral vector. Once a hidden state is computed with BN, a HMM is used in a separate step.

Speaker Identification

DBNs are used in Speaker Recognition in [Sang *et al.*, 2003]. The relationships between the variables is that represented in a coupled HMM and specially in multi-band systems. This topology is chosen because the system is tested in a text-dependent context. In [Arcienega and Drygajlo, 2002] spectral information is used with pitch and voicing status.

4.2 Graph Theory

Universal applicability of graphical models is due to a number of factors. Firstly and most important graphs can visually represent the scientific content of a given model and facilitate communication between researchers and statisticians. Another factor is that these models are naturally modular so that complex problems can be described and handled by a combination of simple elements. And a very important reason is that graphical models are natural data structures for modern digital computers.

Graph Theory [Lauritzen, 1996; Castillo *et al.*, 1997] is just a part of the set theory that works with binary relationships between a countably set and itself. Graph Theory has a rich and specialized framework because of its use in large application fields like physics, economics, telecommunications, chemistry, psychology, etc.

A set is a collection of well defined objects. If the set contains a nonnegative integer number of elements is said to be a finite set. Let S be a finite set, consider all the couples (X_i, X_j) formed with the elements of S, then the set formed by all the couples (X_i, X_j) is called the cross-product set noted by $S \times S$. Now consider the set S and the associated product set $S \times S$, and suppose that some couples have the property **A** while the others have the property $\neg$**A**. This property **A** makes a partition into two parts. If the set $S \times S$ is divided according to the property **A** and if a difference is made between both parts a graph is realized. An example of graph is a binary image where the property, color of pixels (black or white) could be used to build the graph. Another example is a line drawn in a sheet. Let the set $S = \{a, b, c, d\}$ and the subset $B = \{(a, b), (a, d), (b, b), (b, c), (b, d), (c, c), (d, a), (d, b), (d, c), (d, d)\}$ of $S \times S$ be a graph of S. This graph can be depicted as is shown in Figure 4.1.

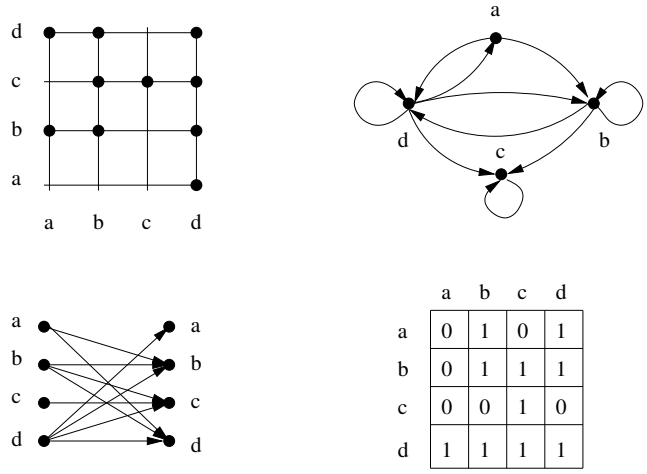


Figure 4.1: *Four different ways to represent a graph. The first uses a grid. The second employs edges and arrows. The third uses coupled points and the last representation is done with an adjacency matrix.*

From Figure 4.1 the last representation is called the adjacency matrix of a simple graph. The matrix with rows and columns labeled by graph vertices has a 1 or 0 in position (i, j) according to whether X_i and X_j are adjacent or not. For a simple graph where the couple (X_i, X_i) is not present, the adjacency matrix must have 0s on the diagonal. The adjacency matrix associated to an undirected graph is symmetric.

Then, for the representation of a graph G it is necessary just a set $S = \{X_1, X_2, \ldots, X_n\}$, a collection of vertices that will be represented by nodes or vertices in the graph, and a set of edges $L = \{L_{i,j} | X_i$ and X_j are connected $\}$ a subset of the set $S \times S$ of ordered pairs of distinct vertices represented by arrows or edges in the graph. Therefore, a graph G is entirely defined by the couple $\{S, L\}$.

A basic feature of a graph is its visual representation. Edges $E \in L$ with both $L_{i,j}$ and $L_{j,i}$ in L are called undirected edges, whereas an edge $L_{i,j}$ with its opposite $L_{j,i}$ not in L is called directed edges. A *line* joining X_i to X_j represents an undirected edge, whereas an *arrow* from X_i pointing to X_j is used to represents a directed edge $L_{i,j}$ with $L_{j,i} \notin L$, see Figure (4.1).

If the graph has only undirected edges it is an undirected graph and if it has only directed edges the graph is said to be a directed graph. A graph can be a mixture of a directed and an undirected graph, called

a mixed graph, but those graphs are out of the scope of this thesis, then they will just be mentioned in some special cases.

4.2.1 Undirected Graphs

The order that defines the edges in an undirected graph has no importance. The undirected graph in Figure (4.2) is completely defined by the couple (S, L), where S is the set of vertices and L the edge set :

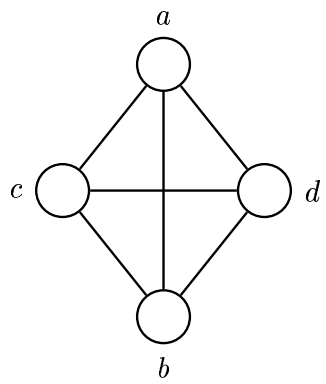


Figure 4.2: *Example of an undirected graph* $\{S, L\}$.

$$S = \{a, b, c, d\}, \tag{4.2}$$

$$L = \{L_{a,b}, L_{b,a}, L_{a,c}, L_{c,a}, L_{a,d}, L_{d,a}, L_{c,b}, L_{b,c}, L_{c,d}, L_{d,c}, L_{b,d}, L_{d,b}\}. \tag{4.3}$$

If there is a line between X_i and X_j, X_i and X_j are said to be adjacent or neighbors. If there is no line between X_i and X_j, i.e. $X_i \neq X_j$, then X_i and X_j are said to be non-adjacent. The set of neighbors of a vertex X_i is denoted as $Ne(X_i)$ and $Ne(A) = \cup_{X_i \in A} Ne(X_i) \setminus A$ denotes the collection of neighbors of vertices in subset A that are not themselves elements of A.

An undirected graph is said to be a complete graph if there is an edge between all pairs of vertices in the graph. For example, the graph in Figure 4.2 is complete because all the nodes in the graph are joined by edges.

If a set A is a subset of the vertex set S, $A \subset S$, this set induces a subgraph $G_A = \{A, L_A\}$, where the edge set $L_A = L \cap (A \times A)$ is obtained from the initial graph G by keeping just the edges with both endpoints in A.

A subset is complete if it induces a *complete subgraph*. From this definition it can be said that all vertices joined by an edge make a complete subgraph. A complete subgraph which is maximal, that is the subgraph is not a subset of any other subset ($\subseteq$), is called a *clique*. In Figure 4.3 the graph defined by the subset $S_{a,b} = \{a, b\}$ is a clique given that both vertices are joined by an edge and $S_{a,b}$ is not contained in any other subgraph. The subgraph $S_{c,d} = \{c, d\}$ can not be a clique because the vertices $\{c, d\}$ are contained in the complete subgraph defined by the vertices $S_{a,c,d} = \{a, c, d\}$.

A path of length n from X_i to X_j is a sequence $X_i = X_0, \ldots, X_n = X_j$ of distinct vertices such that $L_{k-1,k} \in L$ for all $k = 1, \ldots, n$. If there is a path from X_i to X_j it is said that X_i leads to X_j, $X_i \mapsto X_j$.

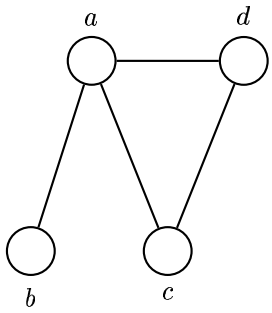


Figure 4.3: *Undirected graph with two cliques. The first formed by the vertices* $\{a, b\}$ *and the second by* $\{a, c, d\}$.

If both $X_i \mapsto X_j$ and $X_j \rightsquigarrow X_i$ it is said that X_i and X_j are connected.

A subset $C \subset S$ is said to be an $(X_i, X_j)-$separator if all paths from X_i to X_j intersect C. Thus, in a undirected graph, C is a $(X_i, X_j)-$separator if and only if :

$$[X_i]_{S\setminus C} \neq [X_j]_{S\setminus C}. \tag{4.4}$$

The subset C is said to separate A from B if it is an $(X_i, X_j)-$separator for every $X_i \in A$ and $X_j \in B$.

An $n-$cycle is a path of length n with the modification that $X_i = X_j$, that means, the path begins and ends in the same vertex.

There are two types of undirected graphs. A graph with at least one path between all couple of vertex is called a connected graph. A tree is also a connected, undirected graph but without cycles. It has a unique path between any two vertices.

4.2.2 Directed Graphs

As already defined, a graph G is a directed graph if all edges in the graph are directed. The undirected version G^u of a graph G is the undirected graph obtained from G by substituting lines for arrows. At the opposite, obtaining a directed graph G^d from an undirected graph G can be done by a given order relation in the vertex set S. But this order has to takes into account all the vertices in S. If this condition is not hold a mixed graph is obtained.

If there exists an arrow from X_i pointing towards X_j, X_i is said to be a parent of X_j and this vertex a child of X_i. The set of parents of X_j is noted as $Pa(X_j)$ and the set of children of X_i as $Ch(X_i)$. For example, in Figure (4.4), vertex a is the parent of vertex b.

In the same way than for undirected graphs, where a line between X_i and X_j, and between X_i and X_j defines two adjacent vertices or neighbors, in a directed graph two vertices are adjacent vertices or neighbors if there is an arrow between these vertices. Conversely, X_i and X_j are said to be non-adjacent if there is no arrow that joins both vertices. The set of neighbors of a vertex X_i is denoted as $Ne(X_i)$.

The expressions $Pa(A) = \cup_{X_i \in A} Pa(X_i)\setminus A$, $Ch(A) = \cup_{X_i \in A} Ch(X_i)\setminus A$, and $Ne(A) = \cup_{X_i \in A} Ne(X_i)\setminus A$ denote respectively the collection of parents, children, and neighbors of vertices in A that are not them-

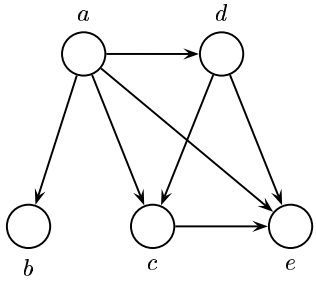


Figure 4.4: *Example of an directed graph.*

selves elements of subset $A \subset S$.

The boundary of a subset of vertices A, $Bd(A)$, is the set of vertices in $S \setminus A$ that are parents or neighbors to vertices in A, then $Bd(A) = Pa(A) \cup Ne(A)$.

The vertices X_i such that $X_i \rightsquigarrow X_j$ and $X_j \rightsquigarrow X_i$ are the ancestors of X_j, $An(X_j)$, and the descendants $De(X_i)$ of X_i are the vertices X_j. The non-descendants are $Nd(X_i) = S \setminus (De(X_i) \cup \{X_i\})$.

A chain of length n from X_i to X_j is a sequence $X_i = X_0, \ldots, X_n = X_j$ of distinct vertices such that X_{k-1} is a parent of X_k or X_k is a parent of X_{k-1} for all $k = 1, \ldots, n$.

An $n-$cycle can have place in a directed graph. An $n-$cycle is said to be directed if it contains an arrow. A very important class of directed graph is that where there are no cycles. This type of graph, called Directed Acyclic Graphs ($DAGs$), is the base graph for the probabilistic models called Bayesian Networks.

A directed graph G is called connected if its G^u graph is a connected graph. A rooted tree is the directed acyclic graph obtained from a tree by choosing a vertex as root and directing all edges away from this root. A forest is an undirected graph where all connected components are trees.

In a chain graph the vertex set S can be partitioned into numbered subsets, forming a dependence chain $S = S(1) \cup \cdots \cup S(T)$ such that all edges between vertices in the same subset are undirected and all edges between different subsets are directed, pointing from the set with lowest number to the one with highest number. Such graphs are characterized by having no directed cycles and connected components forming a partition of the graph into chain components. A graph is a chain graph if and only if its connected components induce undirected subgraphs. The chain components are most easily found by removing all arrows before taking connected components. An undirected graph is a special case of a chain graph. A directed acyclic graph is a chain graph with all chain components consisting of one vertex.

For a chain graph G we define its moral graph G^m as the undirected graph with the same vertices set but X_i and X_j adjacent in G^m if and only if either $X_i \rightarrow X_j$ or $X_j \rightarrow X_i$ or if exists X_k, X_l in the same chain component such that $X_i \rightarrow X_k$ and $X_j \rightarrow X_l$. If no edge have to be added to form the moral graph, the chain graph is said to be perfect.

In the special case of a directed, acyclic graph the moral graph is obtained from the original graph by "marrying parents" with a common child and subsequently deleting directions on all arrows.

A chain component C is said to be terminal if none of the vertices in C have children. A chain graph has always at least one terminal chain component. A terminal component with only one vertex is a terminal vertex.

4.3 Triangulated Graphs

It is a special class of undirected graphs which has many applications in many fields. But in this work the most important application lies in the decomposable graphs which will be defined in the next section.

A chord is an edge that joins two vertices in a cycle and that does not belong to the cycle. A triangulated graph is defined as an undirected graph that has a chord in every cycle of length n ≥ 4. For example, Figure 4.5 shows a triangulated graph. The edge $L_{a,b}$ is a chord for the cycle $\{a, c, b, d\}$. Unlike the graph in Figure 4.5, the graph in Figure 4.6 is not a triangulated graph because the cycle of length equal to 4 $\{a, b, c, d\}$ has no a chord.

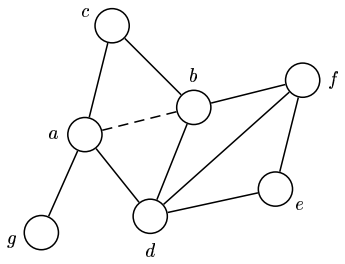


Figure 4.5: *Example of Triangulated Graph. The dashed edge $L_{a,b}$ is a chord for the cycle $\{a, c, b, d\}$.*

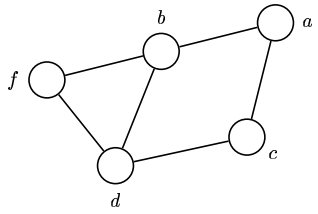


Figure 4.6: *Example of a non Triangulated Graph. The cycle $\{a, b, c, d\}$ has no chord.*

It is easy to see that a non triangulated graph can becomes a triangulated graph by a process of adding chords. For example in the graph in Figure 4.6 a chord $L_{a,d}$ makes the graph a triangulated graph, and the cord $L_{b,c}$ could also be a solution. This fact shows that the triangulation process can give different topologies in the final structure because a cycle can be broken in many ways.

4.4 Graph Decomposition

A triple (A, B, C) of disjoint subsets of the vertex set S of an undirected graph G is said to form a decomposition of G if $S = A \cup B \cup C$ and the two conditions below hold :

(i) C separates A from B;
(ii) C is a complete subgraph of V.

If the graph G verifies those conditions it is said that the sets (A, B, C) decomposes G in components $G_{A\cup C}$ and $G_{B\cup C}$. In this definition the empty sets are allowed. If the sets (A, B, C) are not empty the decomposition is called proper.

A decomposable graph is one that can be successively decomposed into its cliques. And finally, an undirected graph is said to be decomposable if it is complete, or if there exists a proper decomposition (A, B, C) into decomposable subgraphs $G_{A\cup C}$ and $G_{B\cup C}$.

Then from this definition it can be stated that a graph G is decomposable and that G is triangulated are equivalent expressions. And even, an undirected graph is decomposable if and only if it is triangulated.

4.5 Hypergraphs

A hypergraph is a collection $\mathcal{H}$ of subsets of a finite set H, which is the base set. The elements of $\mathcal{H}$ are called hyperedges. In most cases the base set will be the union of hyperedges. A set of complete subsets of a graph G, the cliques $C(G)$ of a graph is a classic hypergraph denoted the clique hypergraph.

One can build a graph by putting together vertices with a characteristic in common. Let G be a graph $G = (S, L)$ and a set $C = \{C_1, C_2, \ldots C_m\}$ obtained from X such that $X = C_1 \cup C_2 \cup \cdots \cup C_m$. The graph $G' = (C, L')$ is called the conglomerated graph of G if all the edges in L' verify $L_{i,j} \in L' \Rightarrow C_i \cap C_j \neq \emptyset$.

If a conglomerated graph associated to an undirected graph G has all the edges that joint the conglomerates with a common vertex the graph is called a junction graph.

For example, the original graph at the left of Figure 4.7 has an associated junction graph represented at the right of Figure 4.7.

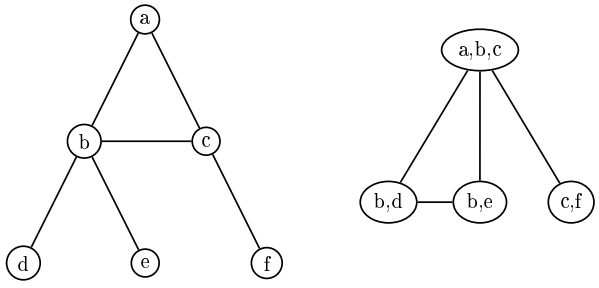


Figure 4.7: *A junction graph (at the right) associated to the original graph in the left*

A decomposable hypergraph $\mathcal{H}$ is a hypergraph which can be obtained by direct joins of hypergraphs that have less hyperedges. And a tree with hyperedges $\mathcal{H}$ as vertices of the tree is called a junction tree in $\mathcal{H}$ if it it is hold that for any two hyperedges a and b in $\mathcal{H}$ and any h on the unique path between a and b $a \cup b \subseteq h$.

Using the same example, the graph at the left side of Figure 4.7, a junction tree can be obtained from the that graph like the one depicted in Figure 4.8.

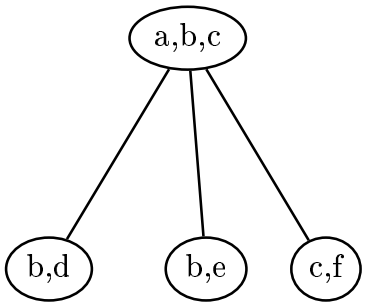


Figure 4.8: *Example of junction tree.*

In Figure 4.8, lets take b in the conglomerate in the bottom left side and b in the bottom middle side. The only path that joins those conglomerates passe by the one in the top of the figure, and this one has also the vertex b.

4.6 Conclusions

This Chapter was intended to provide a sight about Graphical Models, its utilization, some basic concepts and language that will be useful to develop the techniques presented in the next. Their origin was presented at the beginning and some examples of applications in different areas in the following sections. It was show how Graphical models can be applied to problems in administration, biology, queries in the Internet, and particularly in the speech signal processing area. Specially some application of Bayesian Networks in speech recognition as well as in speaker recognition were presented as a first approach to advance the work present in the next Chapters.

Concepts and definition given in this Chapter will help us on the understanding and interpreting of graph presented on next Chapters, which will have as main subject Bayesian Networks. Graphical representations of random variables will be briefly summarized.

Part II

Modeling with Bayesian Networks: Inference, Learning and Adaptation

Chapter 5

Bayesian Networks

A Bayesian Network (BN) [Pearl, 1988; Castillo *et al.*, 1997; Jensen, 1996] is a graphical model representing conditional independencies between a set of random variables. A BN is a couple $(G, CPDs)$ formed by one structure, the graph G, which is a Directed Acyclic Graph (DAG) and a set of Conditional Probability Distributions (CPD), one for each node with parents in the network. For nodes without parents it has just to be specified their prior probability. Consider three variables A, B, and C. From basic probability theory the joint probability can be written as a product of conditional probabilities :

$$P(A, B, C) = P(A)\ P(B|A)\ P(C|A, B). \tag{5.1}$$

Assume that variable B be independent from A. Taking into account this conditional independence the same equation is written as :

$$P(A, B, C) = P(A)\ P(B)\ P(C|A, B). \tag{5.2}$$

A BN is just a graphical way to represent the conditional independencies found in the variables relationships and reflected in the factorization of a joint distribution. For example, the graph in Figure 5.1 where the variable C has two **converging arrows** is used to represent the factorization in equation 5.2.

A directed edge from A to B represents the conditional independence of B given A in the factorization of the joint distribution. Those dependence relations induce a factorization in the joint distribution function expressed as follows for a set of variables $X = \{X_1, \ldots, X_N\}$:

$$P(X) = \prod_{i=1}^{N} P(X_i|Pa(X_i)), \tag{5.3}$$

The semantics of a BN is really easy. Each variable is conditionally independent from its non-descendents given its parents. It has to be noticed that a joint distribution can be factorized in many ways, then there are also many BN consistent with a particular joint distribution.

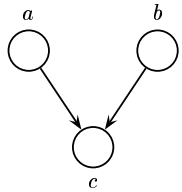


Figure 5.1: *Associated graph to the equation 5.2.*

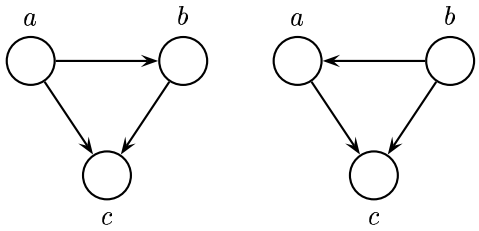


Figure 5.2: *Two equivalent Graphs.*

5.1 Conditional independence

The *d-separation* [Pearl, 1988], where *d* is for direct, is an important concept for the representation on BN. It is said that C *d-separates* A and B if along every undirected path between both set of vertex, or variables in this case, A and B, there is a vertex D such that :

(i) D has converging arrows and neither D nor its descendents are in C, or
(ii) D does not have converging arrows and D is in C.

Two disjoint set of variables A and B are conditionally independent given a set C if an only if C *d-separates* A and B, where *d* is for direct.

The absence of edges in BN implies conditional independencies that can be exploited to obtain better algorithms for computing marginal and conditional probabilities. There are two main research problems in probabilistic reasoning using BN : learning and inference [Murphy, 2002]. BN inference involves computing the posterior marginal probability distribution of some query nodes, and computing the most probable explanation given the values of some observed nodes once the structure is known. The second problem, that will be treated in the third chapter is structure learning.

5.2 Equivalent Graphs

Two graphical models are equivalent if they represent the same set of conditional independencies. The graphical models based on undirected graphs are not redundant because two different graphs always represent two different set of conditional relationships. However, with directed graphs the same conditional probability function can be represented with several graphs.

In order to define the equivalence between two BNs first a V**-structure** has to be defined. A V**-structure** is formed by three vertices $\{a, b, c\}$ in a BN. Those vertices are connected forming a structure with converging arrows, such as in Figure 5.1, where a and c are connected by edges that go from a to c and from b to c.

Two BNs are equivalent [Verma and Pearl, 1991] if :

1. both G^u (the undirected version) are the same and,
2. they share the same V**-structures**.

For example, in Figure 5.2 the probability density function for the graph at the left side is :

$$P(A, B, C) = P(A)\ P(B|A)\ P(C|A, B), \tag{5.4}$$

and for the graph at the right side the probability density function is :

$$P(A, B, C) = P(B)\ P(A|B)\ P(C|A, B). \tag{5.5}$$

Using the Bayes rule the next equation is obtained :

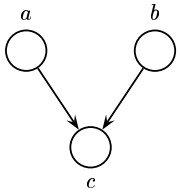
$$P(B)\ P(A|B) = P(A)\ P(B|A), \tag{5.6}$$

then, both graphs are equivalent.

5.3 Multinomial BN

In a multinomial BN all the variables $\{X_i\}$ are discrete. Thus, the conditional probability function associated to each variable $\{X_i\}$ is a multinomial probability function. This type of probability function can be defined either in a numerical or parametric way. The numbers or parameters are given in a table (Conditional Probability Table **CPT**) for each possible combination of values taken by the variables.

For example, if all the variables are binary $X = \{x, \neg x\}$, the parameters (CPT) for the graph in Figure 5.1 could be those specified in the next tables :



a	$p(a)$
0	0.25
1	0.75

b	$p(b)$
0	0.2
1	0.8

a	b	$p(c\|a,b)$
0	0	0.25
0	1	0.50
1	0	0.75
1	1	0.50

5.4 Multinormal BN

In a gaussian or normal BN all the variables are modeled by a normal distribution $\mathcal{N}(x; \mu, \Sigma)$. This kind of BN are also called Gaussian Networks (GN) [Shachter and Kenley, 1989] leaving the name BN for networks with discrete variables. The normal distribution is given by the equation 2.3 and repeated here:

$$f_{\mathcal{N}}(x) \backsim \mathcal{N}(x; \mu, \Sigma) = \frac{1}{(2\pi)^{-\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}. \tag{5.7}$$

where μ is the d-dimensional mean vector, Σ is the associated $d \times d$ covariance matrix, $|\Sigma|$ the determinant of Σ and $(x - \mu)^T$ is the transpose of $(x - \mu)$.

The density function for each factor in equation 5.3 of the BN is defined by a product of conditional probability functions [Shachter and Kenley, 1989] as :

$$f(x_i|Pa(x_i)) \backsim \mathcal{N}(x; \mu_i + \sum_{j=1}^{i-1} \beta_{i,j}(x_j - \mu_j), \nu_i), \tag{5.8}$$

where $\beta_{i,j}$ is the regression coefficient of x_i and $Pa(x_i)$. $\mathcal{N}(x; \mu, \nu)$ is a univariate normal distribution. Given this form, a value of zero for any $\beta_{i,j}$ parameter implies that there are not an arc from X_i to X_j. The parameters of this model are as follows. μ_i is the unconditional mean value of X_i, ν_i is its the conditional

variance given its the parents:

$$\nu_i = \Sigma_i - \Sigma_{iPa_i}\Sigma_{Pa_i}^{-1}\Sigma_{iPa_i}^T \tag{5.9}$$

where the conditional variance of x_i given $Pa(x_i)$. Σ_i is the non conditional covariance, Σ_{iPa_i} the covariance between x_i and $Pa(x_i)$ and Σ_{Pa_i} the variance of $Pa(x_i)$. Therefore $\beta_{i,j}$ measures the relation between x_i and $Pa(x_i)$. For example, if just two continuous variables are used in a BN, figure 5.3.

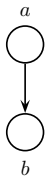


Figure 5.3: *Two continuous variables used dnas le example of normal distribution with BN.*

If $\{a, b\}$ are both continuous variables:

$$f_a \sim \mathcal{N}(a; \mu_a, 1/\nu_a),$$
$$f_b \sim \mathcal{N}(b; \mu_b, 1/\nu_b),$$

then, the density function is the linear-regression model written in the following equation:

$$f(a,b) = \frac{1}{\sqrt{2\ \pi\ \nu_a}} \exp -\frac{1}{2}\left(\frac{a-\mu_a}{\nu_a}\right)^2 \frac{1}{\sqrt{2\ \pi\ \nu_b}} \exp -\frac{1}{2}\left(\frac{b-(\mu_b+\beta_{a,b}(a-\mu_a))}{\nu_b}\right)^2, \tag{5.10}$$

where $\beta_{a,b}$ measure the relation between both variables.

5.5 BN and GMM

Mixture Models are a type of density models which comprise a number of components. These components are combined to provide a multimodal density. A Gaussian Mixture Model (GMM) is defined as a combination of gaussian densities. A Gaussian density in a d-dimensional space, characterized by its mean $\mu \in \mathbb{R}^d$ and a $d \times d$ covariance matrix Σ, is defined in equation 5.7 and repeated here :

$$f_{\mathcal{N}}(x) \backsim \mathcal{N}(x; \mu, \Sigma) = \frac{1}{(2\pi)^{-\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}.$$

The conditional density for a vector **x** given a M components GMM is defined by $\Lambda = \{w, \mu, \Sigma, \}$:

$$p(x|\Lambda) = \sum_{i=1}^{M} w_i f_{\mathcal{N}}(x), \tag{5.11}$$

where $\sum_{i=1}^{M} w_i = 1$ and $0 \leq w_i \leq 1\ \forall i$.

The mixing parameter w_i corresponds to the prior probability that x was generated by the component i.

Now, Figure 5.4 represents a BN with two vertices. The first vertex A, represents a M states discrete variable which verifies $\sum_{a_i} P(A = a_i) = 1$ and $0 \leq P(A = a_i) \leq 1$. The second vertex B represents a variable which follows a gaussian distribution conditioned to the value taken by the first variable $A = a_i$.

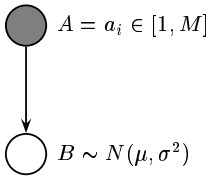


Figure 5.4: *GMM represented with a BN.*

Using the conditional independencies reflected in the structure, the conditional probability can be written as :

$$P(A, B) = P(A)P(B|A), \tag{5.12}$$

and the conditional density for a vector **b** is written as :

$$p(B = b) = \sum_{i=1}^{M} p(B = b|A = a_i)\, p(A = a_i). \tag{5.13}$$

where :

$$\begin{cases} p(A = a_i) = w_i, \\ p(B = b|A = a_i) = \mathcal{N}(b; \mu_i, \Sigma_i). \end{cases} \tag{5.14}$$

5.6 Dynamic Bayesian Networks

Time is a very important variable in almost all the events in the real life. The time evolution of variables is a field generally known as time series analysis. Dynamic Bayesian Networks (DBNs) describe a system that is changing or evolving in time using the formalism of BN. This kind of representation allows to model (predict, filter, smooth, update, monitor) systems which depend on time.

Usually, unlike temporal models which just give an idea about changes in the variables values in DBN, dynamics is related to systems which change not just the values of variables over time, but also the dependence between those variables. Hence, temporal models would be a sub-class of dynamic ones. If every time slice of a temporal model corresponds to one particular state of a system, and if the movement between the slices reflects a change in the state instead of time, that model is classified as a dynamic model. Then, even if the concerned systems are called dynamic in reality they are temporal all along this work. That is, relations between variables are the same along the time.

5.6.1 Definition

DBNs are defined as special case of singly connected BN specifically aimed at time series modeling. All vertices, edges and probabilities that form static interpretation of a system are identical to a BN. Also, the states of a system described as a DBN satisfy the Markovian condition; the state of a system at time t knowing its full evolution until time T depends only on its immediate past. Then, generally, in DBN states of a system at time t may depend on states at time $t - 1$ and possibly on current states of some other nodes in the time slice.

A DBN consists of probability distribution function on the sequence of T hidden $H = \{h_0, \ldots, h_{T-1}\}$ and T observed variables $O = \{o_0, \ldots, o_{T-1}\}$, where T is the time index. If all the hidden and observed variables are in the same variables set $X = \{H, O\}$, the semantics of a DBN can be defined by "unrolling" a BN until having T time slices. The resulting joint distribution can be written as

$$P(x_{1:T}) = \prod_{t=1}^{T} \prod_{i=1}^{N} P(x_i(t)|Pa(x_i(t))), \tag{5.15}$$

where N is the number of variables for each time slice. Note that parents of $x_i(t)$ can be into the present slice or into the past slice.

5.6.2 Hidden Markov Models as DBN

Hidden Markov Models (HMM) are an example of a graphical model where exact inference is tractable. The difference between a DBN and a HMM is that in a HMM the state space consists of a single random variable X_t and in a DBN hidden states are represented in terms of a set of random variables $X_1, \ldots, X_T$.

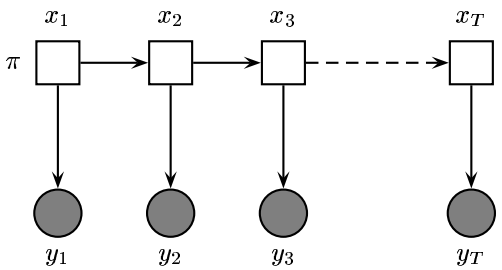


Figure 5.5: *A HMM represented as a DBN.*

A HMM is a graphical model in form of a chain, see Figure 5.5. In this model, the sequence of multinomial state nodes x_i is assumed to verify the Markov property. The conditional probability of a node x_i, given its immediate predecessor x_{i-1} is independent of all other previous variables. Also, in this model, the state chain is supposed to be homogeneous, that is, the matrix of transition probability, $A = P(x_i|x_{i-1})$, is invariant across time. It is also necessary to know the prior probability distribution $\pi = P(x_1)$ for the initial state x_1. And finally, for the observed nodes Y_i time invariant emission probability law $B = P(y_i|x_i)$ is also given.

In HMM the output nodes are treated as evidence and the state nodes as hidden nodes for training. Usually, the Expectation-Maximization (**EM**) [Dempster *et al.*, 1997] algorithm is used to update parameters $\lambda = \{A, B, \pi\}$. In the first step an inference algorithm computes the conditional probabilities $P(x_i|\{y_i\})$ and $P(x_i, x_{i-1}|\{y_i\})$. In the second step the parameters λ are updated via weighted maximum likelihood with the weights obtained in the first step using the the conditional probabilities.

Now, it can be seen, that exact inference in HMM is tractable, because the cliques size is small (equal to 2) . The moralization and triangulation step are not necessary for HMM.

5.6.3 Factorial Hidden Markov Models

It could be thought that HMMs are limited in structure. May be, some additional structural assumptions about the state space and the transition probabilities can be helpful in some problems. That is the reason why structured versions of HMM have been studied recently. For example the coupled HMM that will be used in the next chapter. Another example is the factorial HMM [Ghahramani and Jordan, 1997], Figure 5.6.

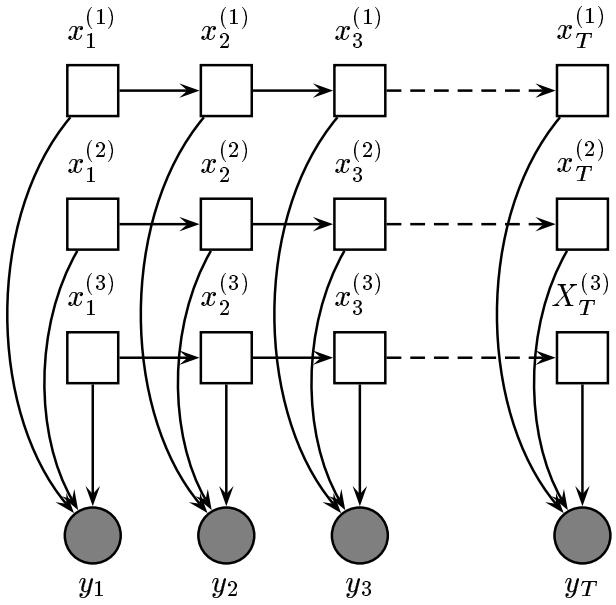


Figure 5.6: *A Factorial HMM represented as a graphical model.*

The system is composed of a set of M chains. Each variable $x_i^{(m)}$ is the hidden state at the time slice i and in the chain $m \in M$. In this model there are M state transition matrix A^m, one for each chain. The total state space in this model can be seen as the Cartesian product of the state space in each individual chain. The overall transition probability for the system by taking the product across the intra-chain transition probabilities is :

$$P(x_i|x_{i-1}) = \prod_{m=1}^{M} A^{(m)} P(x_i^{(m)}|x_{i-1}^{(m)}), \tag{5.16}$$

The Factorial HMM is a model for systems in which hidden states are realized from an uncoupled set of dynamical systems and with an only available observation.

Inference in such model is not complex. First, after the moralization and triangulation the undirected graph for $M = 2$ is shown in Figure 5.7. The advantage of this model is that it represents a large effective state space with a much smaller number of parameters than a single HMM.

Cliques in this model are of size equal to three. Thus time complexity of exact inference is most higher than inference in a simple HMM. If the number of chains increases the complexity is also augmented.

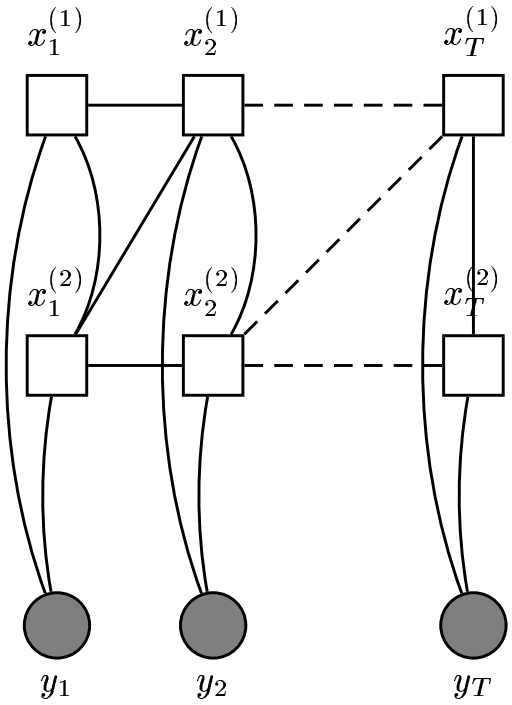


Figure 5.7: *A Factorial HMM after moralization and triangulation with* $M = 2$.

5.6.4 Coupled HMM

In the coupled HMM [Brand, 1996] the hidden variables in each HMM interact locally with their neighbors and each hidden variable has its own observed variable, as shown in Figure 5.8 for a coupled HMM with two chains. This models have several applications. For example, using two HMMs, one of them can represents the visual flow and the second one the audio flow in audio visual speech recognition [Nefian *et al.*, 2002]. Another example is the Multi-band Speech Recognition [Daoudi *et al.*, 2003]. Each HMM represents the temporal dynamics of the selected band and the relation between the hidden states represents the frequency dynamics.

5.7 Conclusions

In this Chapter we presented the principles of Bayesian Networks and Dynamic Bayesian Networks. Conditional independence and equivalence between graphs concepts were also presented. It has described Multinomial and Multinormal distributions representation using Bayesian Networks. This representation introduce the statistical analysis researched in the all this work. This Chapter let already see how different source of information can be combined in a unique statistical model.

Next Chapters will present the basic problems found in Graphical Models, that is inference and learning. They develop the concept of modeling in the module of our SV system.

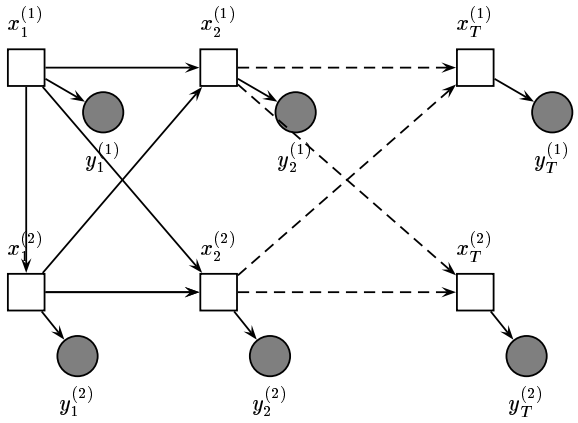


Figure 5.8: *A Coupled HMM with two main HMMs.*

Chapter 6

Inference

In this chapter the inference problem [Murphy, 2002] is treated. It is wanted to know how the conditional probabilities of one or several variables X in the network changes given that some others $Y = y$ has been observed, that is, we search for the next probability $P(X|Y = y)$. Some exact as well as approached techniques will be described. This chapter is important because it gives the bases used for parameter learning with hidden variables.

As mentioned above, evidence, i.e. knowledge about the state of one or several variables, would modify the probability of other variables in the network. Doing probabilistic inference consists in computing the probability of each state of a variable when we know the state taken by some other variables, that is, the actualization of the probability of variables as a function of evidence. This process is called marginalization in classic probability. Actually the problem of inference is not just a theoric problem. Let's take the example of DFT computation. The FFT transformation is just a faster way to compute the DFT who takes advantage of symmetric properties of DFT. As well as FFT, inference techniques in BN take advantages of conditional independencies and structure topology in order to compute the probability of each state in a tractable and faster way. The first works [Pearl, 1988; Kim and Pearl, 1983] about this problem propose some mechanisms of inference in graphical models that work in tree or forest structures. Some time later [Lauritzen and Spiegelhalter, 1988] a method of inference propagation was proposed which works in DAGs without the restriction of tree structures. This method is based on an associated structure called a clique tree or junction tree (section 4.5).

6.1 Exact Inference

The simplest way to solve the problem of propagation is summing out all the variables in the network from the joint probability distribution as it was mentioned above. However, this procedure is inefficient since some computations are made several times. Another reason to reject this procedure is the necessary memory space which could be huge. For example, if all the variables are binary $(x, \neg x)$, 2^n locations is the necessary memory space for a network with n variables. Then, it can be seen that bigger network needs much more memory.

6.2 Variable Elimination

The idea here is to establish an order in which the query variable x_i is the last in it. At each step, one of the variables will be eliminated combining all the factors where it is present and summing out. After this step one variable of those is eliminated and the whole function does not depend anymore on it. At the end, a potential is obtained which is proportional to the searched posterior probability.

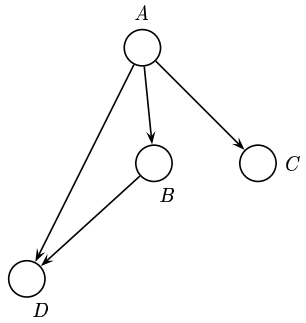


Figure 6.1: *BN for the variable elimination example.*

For example, consider the BN in Figure 6.1. The joint probability distribution is :

$$P(A, B, C, D) = P(A)\ P(B|A)\ P(C|A)\ P(D|A, B). \tag{6.1}$$

The probability $P(A)$ can be computed summing out the variables B, C, D. This equation can be written as follows:

$$\begin{aligned} P(A) &= \sum_{B,C,D} P(A, B, C, D) \\ &= P(A) \sum_{B} P(B|A) \sum_{D} P(D|A, B) \sum_{C} P(C|A). \end{aligned} \tag{6.2}$$

Of course each term $\sum_X P(X|Y)$ is equal to 1 for all Y, but this is not always the case. For example, if X takes a value, that is, if X is observed the sum is just over the observed state of the variable. In this equation is already reflected the computational reduction effect obtained with the only clever position of sums. In this way the computational work is minimized because the variables at the right side are marginalized and then the final term depends only on one variable.

This way of doing inference shows already the message passing concept. The last term in equation (6.2) $P(C|A)$ after been summed out $\sum_C P(C|A)$ depends only on variable A. Later this result will be marginalized in the sum over A. This process could be seen as a message passed from the term $\sum_C P(C|A)$ to the term $\sum_A P(A)$. In this process a structure is discovered and shown in Figure 6.2. In this Figure the structure obtained given the order in equation (6.2) is presented. The messages $\Upsilon_{X \to Y}$ represent the marginal of term in X that goes to the term in Y.

6.3 Message Passing in Polytrees

Algorithms used for tree structures are the base for more general and real structures, even if BNs issue of problems in the real life have not always this form. The local conditioning algorithm [Díez, 1996] is an example. If the structure is already a tree the inference can be implemented in a message passing procedure as is described in this section.

Between two different nodes in a polytree there is only one path. This important property imply that each node separates the polytree in two disconnected polytrees. One of them contain its parents and all the other nodes connected through its parent. The other polytree contains the children and the nodes connected to it through its children. Therefore the evidence **E=e** is propagated to each node X either by its parents or by its children in such a way that it can be separated into two disjoint sets:

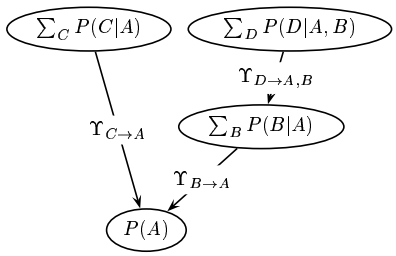


Figure 6.2: *Message passing in the structure obtained with a given order.*

$$\mathbf{E} = \mathbf{e}_X^+ \cup \mathbf{e}_X^-,$$
$$\mathbf{e}_X^+ \cap \mathbf{e}_X^- = \emptyset,$$

where $\mathbf{e}_X^+$ is used to specify the evidence arriving through the parents of X and $\mathbf{e}_X^-$ specify that arriving through its children.

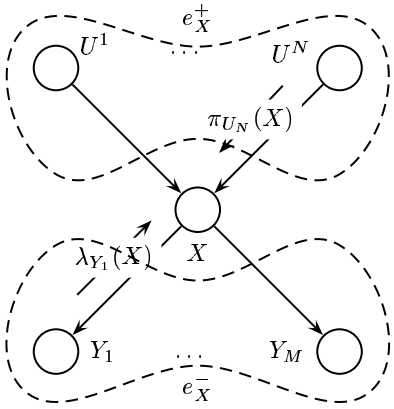


Figure 6.3: *Message Passing in Polytree Structures.*

Then, from Bayes rule, the probability of variable X given the evidence $\mathbf{E} = \mathbf{e}$ is written as follows:

$$\begin{aligned} P(X|\mathbf{E}=\mathbf{e}) &= P(X|\mathbf{e}_X^+, \mathbf{e}_X^-) \\ &= \frac{1}{P(\mathbf{e}_X^+, \mathbf{e}_X^-)}\, P(\mathbf{e}_X^+, \mathbf{e}_X^-|X)\, P(X). \end{aligned} \tag{6.3}$$

Given that X d-separates $\mathbf{e}_X^+$ from $\mathbf{e}_X^-$ in a polytree structure the previous equation can be written as:

$$\begin{aligned} P(X|\mathbf{E}=\mathbf{e}) &= \frac{1}{P(\mathbf{e}_X^+, \mathbf{e}_X^-)}\, P(\mathbf{e}_X^-|X)\, P(\mathbf{e}_X^+|X)\, P(X) \\ &= \frac{1}{P(\mathbf{e}_X^+, \mathbf{e}_X^-)}\, P(\mathbf{e}_X^-|X)\, P(\mathbf{e}_X^+, X) \\ &= \alpha\, P(\mathbf{e}_X^-|X)\, P(\mathbf{e}_X^+, X), \end{aligned} \tag{6.4}$$

where $\alpha = 1/P(\mathbf{e}_X^+, \mathbf{e}_X^-)$ is a normalization constant. From equation 6.4 we can define two terms. The first one, which takes into account the evidence that comes through the children of X is defined as follows:

$$\lambda(X) = P(\mathbf{e}_X^-|X), \tag{6.5}$$

and the second, which takes into account the evidence that comes through the parents of X is:

$$\pi(X) = P(\mathbf{e}_X^+, X). \tag{6.6}$$

Therefore, the conditional probability of X given the evidence can be written in the following form:

$$P(X|\mathbf{E}{=}\mathbf{e}) = \alpha\ \lambda(X)\ \pi(X). \tag{6.7}$$

To compute $\lambda(X)$ and $\pi(X)$ we consider the structure present in Figure (6.3), where the N parents of X are designated by $\mathbf{U} = \{U_1, \ldots, U_N\}$ and its M children by $Y = \{Y_1, \ldots, Y_M\}$. Consequently the evidence $\mathbf{e}_X^-$ and $\mathbf{e}_X^+$ can be decomposed into:

$$\mathbf{e}_X^+ = \{\mathbf{e}^+_{U_1\ X}, \ldots, \mathbf{e}^+_{U_N\ X}\}, \tag{6.8}$$

$$\mathbf{e}_X^- = \{\mathbf{e}^-_{X\ Y_1}, \ldots, \mathbf{e}^-_{X\ Y_M}\}, \tag{6.9}$$

where, $\mathbf{e}^-_{X\ Y_i}$ is the evidence in the sub polytree with root Y_i and $\mathbf{e}^+_{U_j\ X}$ is the evidence in the sub polytree with leaf U_j.

Using those definitions, the value of $\lambda(X)$ is computed as follows:

$$\begin{aligned}\lambda(X) &= P(\mathbf{e}_X^-|X), \\ &= P(\mathbf{e}^-_{X\ Y_1}, \ldots, \mathbf{e}^-_{X\ Y_M}|X).\end{aligned}$$

Since each $\mathbf{e}^-_{X\ Y_i}$ is independent of $\mathbf{e}^-_{X\ Y_j}$ given X for $i \neq j$ in a polytree, then, the last equation can be written as follows:

$$\begin{aligned}\lambda(X) &= P(\mathbf{e}^-_{X\ Y_1}|X) \ldots P(\mathbf{e}^-_{X\ Y_M}|X) \\ &= \prod_{i=1}^{M} P(\mathbf{e}^-_{X\ Y_i}|X).\end{aligned}$$

Defining the message received by X from each one of its children Y_i as $P(\mathbf{e}^-_{X\ Y_i}|X) = \lambda_{Y_i}(X)$ it can be written the next equation:

$$\lambda(X) = \prod_{i=1}^{M} \lambda_{Y_i}(X) \tag{6.10}$$

The term $\pi(X)$ is computed as follows:

$$\begin{aligned}\pi(X) &\equiv P(X, \mathbf{e}_X^+) \\ &= \sum_{\mathbf{U}=\mathbf{u}} P(X, \mathbf{U} = \mathbf{u}, \mathbf{e}_X^+) \\ &= \sum_{\mathbf{U}=\mathbf{u}} P(X|\mathbf{U} = \mathbf{u}, \mathbf{e}_X^+)\ P(\mathbf{U} = \mathbf{u}, \mathbf{e}_X^+)\end{aligned} \tag{6.11}$$

Given that the parents of X have no common ancestors, because of the polytree structure, its parents and its ancestors are d-separated.Therefore, the second term in equation 6.11 can be written as follows:

$$\begin{aligned} P(\mathbf{U}=\mathbf{u}, \mathbf{e}_X^+) &= P(u_1, e^+_{U_1X}, \ldots, u_N, e^+_{U_NX}) \\ &= P(u_1, e^+_{U_1X}) \ \ldots \ P(u_N, e^+_{U_NX}) \\ &= \prod_{i=1}^{N} P(u_i, e^+_{U_iX}) \end{aligned}$$

Defining $P(u_i, e^+_{U_iX}) = \pi_X(u_i)$ the message send by each u_i to X, it can be written the next equation:

$$P(\mathbf{U}=\mathbf{u}, \mathbf{e}_X^+) = \prod_{i=1}^{N} \pi_X(u_i) \tag{6.12}$$

Finally, the value of $\pi(X)$ is computed as follows using the equations (6.11) and (6.12):

$$\pi(X) = \sum_{\mathbf{U}=\mathbf{u}} P(X|\mathbf{U}=\mathbf{u}, \mathbf{e}_X^+) \ \prod_{i=1}^{N} \pi_X(u_i). \tag{6.13}$$

Using this last equation (6.13) and equation (6.10) equation 6.7 can be rewritten in the following form:

$$P(X|\mathbf{E}=\mathbf{e}) = \alpha \prod_{i=1}^{M} \lambda_{Y_i}(X) \left[\sum_{\mathbf{U}=\mathbf{u}} P(X|\mathbf{U}=\mathbf{u}, \mathbf{e}_X^+) \ \prod_{i=1}^{N} \pi_X(u_i) \right] \tag{6.14}$$

Once the variable X has received the messages from its parents and children it can send its own messages to them. Given the equivalence in the structure for all the nodes in the networks the computation of those messages are all in the same relative relation to the query variable, is just a notation formalism, see figure 6.4.

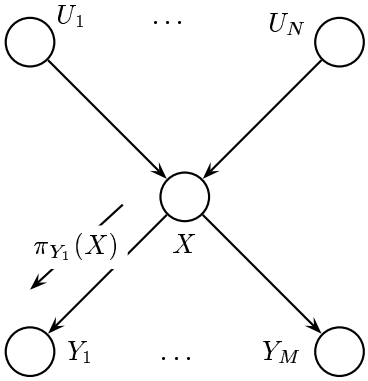


Figure 6.4: *Message Passing in Tree Structures. π computation procedure.*

Let's take the arc XY_j and variable Y_j, in figure 6.4. Evidence arriving to Y_j is divided into $\mathbf{e}^-_{XY_j}$ and $\mathbf{e}^+_{XY_j}$. The second term $\mathbf{e}^+_{XY_j}$ can be decomposed into several sets. In one hand the evidence which comes

from variable X and all the nodes above it e^+_X. In the other hand the evidence $e^-_{X\ Y_k}$ which comes from the children of X, the siblings Y_k of Y_j, where $k \neq j$.

$$\mathbf{e}^+_{XY_j} = \{e^+_X, \bigcup_{k\neq j} e^-_{X\ Y_k}\}$$

Using the **d-separation** X divide $\mathbf{e}^+_X$ from $\bigcup_{k\neq j} \mathbf{e}^-_{XY_k}$ and also the subsets $\bigcup_{k\neq j} \mathbf{e}^-_{XY_k}$ itself. Finally the contribution of X to the node Y_j can be written as :

$$\begin{aligned} \pi_{Y_j}(X) &= P(X, \mathbf{e}^+_{XY_j}) \\ &= \prod_{k\neq j} P(X, \mathbf{e}^+_X, \mathbf{e}^-_{XY_k}) \end{aligned} \tag{6.15}$$

Given that X d-separate $\mathbf{e}^-_{XY_k}$ and $\mathbf{e}^+_X$ next equation is obtained, where all the siblings of Y_i have been taken into account:

$$\begin{aligned} \pi_{Y_j}(X) &= P(X, \mathbf{e}^+_X) \prod_{k\neq j} P(\mathbf{e}^-_{XY_k}|X) \\ &= \pi(X) \prod_{k\neq j} \lambda_{Y_k}(X) \end{aligned} \tag{6.16}$$

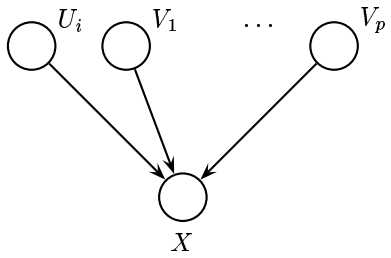


Figure 6.5: *Contribution of parents of X to computation of π. Each variable $\{U_i, V_1, \ldots, V_p\}$ send a message to X*

To finish, let us compute the message sent by X to one of its parents U_i, that is $\lambda_X(U_i) = P(\mathbf{e}^-_{U_i\ X}|U_i)$. See Figure 6.5, where $\mathbf{V}$ represents the set $\{V_1, V_2, \ldots, V_p\}$ of ancestors of X which are different of U_i. Then, the evidence $\mathbf{e}^-_{XY_j}$ is :

$$\begin{aligned} \mathbf{e}^-_{U_i\ X} &= \mathbf{e}^-_X \cup \mathbf{e}^+_{V_1\ X} \cup \mathbf{e}^+_{V_2\ X} \cup \cdots \cup \mathbf{e}^+_{V_p\ X} \\ &= \mathbf{e}^-_X \cup \mathbf{e}^+_{\mathbf{V}\ X} \end{aligned} \tag{6.17}$$

Then, the searched probability is computed as follows:

$$\begin{aligned} \lambda_X(U_i) &= P(e^-_{U_i\ X}|U_i) \\ &= \sum_X \sum_{\mathbf{V}=\mathbf{v}} P(e^-_X, e^+_{\mathbf{V}\ X}, X, \mathbf{V}=\mathbf{v}|U_i) \\ &= \sum_X \sum_{\mathbf{V}=\mathbf{v}} P(e^-_X|\mathbf{V}=\mathbf{v}, e^+_{\mathbf{V}\ X}, X, U_i) P(X|\mathbf{V}=\mathbf{v}, e^+_{\mathbf{V}\ X}, U_i) P(e^+_{\mathbf{V}\ X}, \mathbf{V}=\mathbf{v}|U_i), \end{aligned}$$

From **d-separation** it is known that X separates $\mathbf{e}_X^-$ from nodes above it. Also, it is known that the parents of X are marginally independents. Then, we can write the last equation in the following form:

$$\lambda_X(U_i) = \sum_X \sum_{\mathbf{V}=\mathbf{v}} P(e_X^-|X)P(X|\mathbf{V}=\mathbf{v})P(e^+_{\mathbf{V}\,X}, \mathbf{V}=\mathbf{v}|U_i), \tag{6.18}$$

The last term can be written as follows because of the marginal independence between the parents of X:

$$\begin{aligned} P(e^+_{\mathbf{V}\,X}, \mathbf{V}=\mathbf{v}|U_i) &= P(\mathbf{e}^+_{\mathbf{V}\,X}, \mathbf{V}=\mathbf{v}), \\ &= \prod_{i=1}^{p} P(e^+_{V_l\,X}, v_l) \\ &= \prod_{i=1}^{p} \pi_X(v_l). \end{aligned} \tag{6.19}$$

Then, we obtain finally the next equation:

$$\lambda_X(U_i) = \sum_X \lambda(X) \left[\sum_{\mathbf{V}=\mathbf{v}} P(X|\mathbf{V}=\mathbf{v}, U_i) \prod_{i=1}^{p} \pi_X(v_l) \right]. \tag{6.20}$$

Before finishing with this section some practical comments are given. As it can be seen the procedures are recursive. $\pi(X)$ is computed from $\pi_X(u_i)$, $\pi_{Y_j}(X)$ from $\pi(X)$ and $\lambda_{Y_k}(X)$.

Another point is how to take into account the evidence in this procedure, the initial conditions. For a node $\mathbf{U}$ without parents, $e^+_{\mathbf{U}} = \emptyset$, then $\pi(X) = P(u)$.

For a terminal node Y, $\lambda(X)$ is needed . If any information is available for this node, the same value is given for all y_i, for example $\lambda(y) = 1$. If a value of Y is known, for example y_0, a positive value is assigned to it and 0 to the other values of Y, that is :

$$\begin{aligned} \lambda(y_0) &= 1 \\ \lambda(y) &= 0 \quad \forall \; y \neq y_0 \end{aligned}$$

6.4 Junction Tree

The junction tree algorithm works with an undirected graphical model which is obtained from a directed graphical model. Once the undirected graphical model is obtained, inference calculations are done with the formalism of undirected graphs. In general, the final graph is a tree formed of cliques.

The first step that converts the directed graph into an undirected graph is moralization. As mentioned before (section 4.2.2) moralization in a BN is just the process of "marrying parents" with a common child and subsequently deleting directions on all arrows to obtain an undirected graph, (Figure 6.6).

To understand moralization it has to be noticed that parents are correlated given their children and that in both the directed and undirected cases, the joint probability distribution is obtained as a product of local functions. Then, moralization is the way to preserve such correlation in graphical representation. Without moralization the problem is that these correlated variables do not always appear together within a clique. Therefore, a moral graph represents the probability distribution on the original directed graph within the undirected formalism.

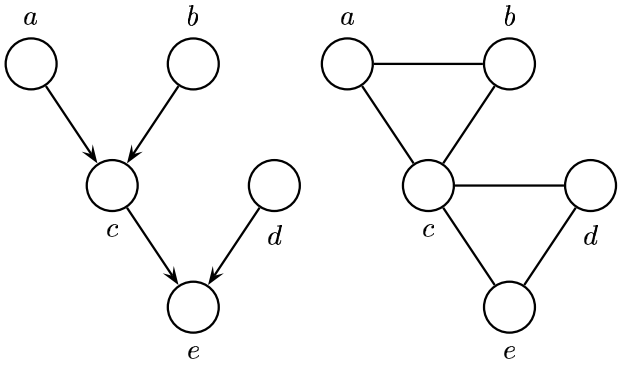


Figure 6.6: *Junction tree Construction. Left side the original graph G. Right side the moral graph G^m.*

The second step, if necessary, in the junction tree construction is triangulation. This process takes a moral graph as input and gives an undirected graph with (or not) extra edges, see section 4.3.

Once a graph is triangulated it is possible to arrange the cliques (section 4.2.1) of the graph into a structure known as **junction tree** (Figure 6.7). A junction tree verifies the **running intersection property**, that is, if a node appears in any two cliques in the tree, it appears also in all the cliques that lie on the path between the two cliques. This property allows inference to be based on local computation because local consistency implies global consistency.

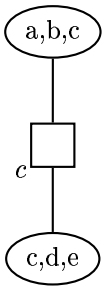


Figure 6.7: *Junction tree obtained from the graph in Figure 6.6. The square node represents the separator and the oval nodes the cliques.*

A **potential** ϕ_x is a function on the set of configurations of a clique which associates a positive real number to each configuration. The configuration of cliques is the possible values taken for all of the nodes in the clique. Then, the probabilistic computation performed on the junction tree involves marginalizing and rescaling the clique potential in order to achieve local consistency between neighboring cliques. This computation uses also the separator potential ϕ_s. A separator is a set that joints the common variables between the cliques that it separates.

Once the junction tree is build, the procedure [Huang and Darwiche, 1994] for making inferences has several steps. The first step is **Initialization**. A potential is assigned to each clique ϕ_{c_i}. This clique potential

depends on the variables potential in that clique in such a way that the final joint distribution verifies :

$$P(U) = \frac{\prod_i \phi_{X_i}}{\prod_j \phi_{S_j}}. \tag{6.21}$$

This joint distribution is associated with an inconsistent junction tree. Generally separator potentials are initialized to the one $\phi_{S_i} \leftarrow 1$. The variables potential ϕ_{X_i} are assigned only to one clique, the one which includes its parents.

Then, the second step is **Propagation**, in which the local computations called *message passing* are done. The message passing process rearranges the junction tree potentials in such a way that the junction tree becomes locally consistent, and then globally consistent.

The message passing between two neighboring cliques C_X and C_Y separated by the separator $S_{X,Y}$ is performed as follows. A message passing from C_X to C_Y occurs in two steps. The first, called Projection, assigns a new potential to $S_{X,Y}$, saving first the old one :

$$\phi_S^{old} \leftarrow \phi_S.$$

$$\phi_S \leftarrow \sum_{X \setminus S} \phi_X. \tag{6.22}$$

then, the second step is **Absorption** in which a new potential is assigned to Y using both the new and the old potentials :

$$\phi_Y \leftarrow \phi_Y \frac{\phi_S}{\phi_S^{old}}. \tag{6.23}$$

Given a junction tree with n cliques, the global propagation starts by choosing a clique, and then, performing $2(n-1)$ message passing. First in the **Collect-Phase** each clique passes its message to the selected clique, starting from the farthest one. In the **Distribute-Phase** the selected clique passes its message to the other cliques.

6.5 Approximate Inference

Given the intractability of exact inference methods in large and multi-connected BN it is important to consider approximate inference algorithms . In general exact methods present some problems. For example, some of them are not applicable to all types of structures and, when the number of nodes and complexity grow methods of general validity become very inefficient. This is not surprising since it has been demonstrated that the exact propagation task is NP-hard [Cooper, 1989]. For that reason, and from a practical point of view, exact propagation methods can be very restrictive and even inefficient in situations for which the type of structure of the network requires a large memory and a lot of computational power. Approximated methods are also NP-hard [Dagum and Laby, 93], but the problems that can be treated with are more extended.

A first approach to approximate inference is to apply the exact inference techniques to general graphs. But, most of the approximated inference methods are based on sampling techniques using Monte Carlo techniques which provide approximate answers whose accuracy depends on the number of samples generated. The problem of this kind of techniques is to obtain those samples from a probability distribution which is hard to handle. To overcome this problem different process have been proposed in the literature. Methods based on importance sampling [Henrion, 1988], on stratified simulation [Buckaert, 1994] and also

on Markov Chains [Pearl, 1987b]. In general, the Monte Carlo algorithms just use the available local information for simulating the value of a given variable.

In addition to those methods, other methods have been proposed based on the idea of simplifying the problem either by simplifying the structure [Kjaerulff, 1993] or by reducing the probability distribution [Jensen and Andersen, 1990].

6.5.1 Loopy Belief Propagation (BLP)

This techniques follows basically the already mentioned techniques in polytrees. A first approach is to create the associated junction tree and then apply the message passing technique on it. This way of proceed has the drawback of clique size. The most big the clique is the most the time consuming the method becomes. A really interesting application of this technique is the error correcting codes found in "Turbo Codes" [McEliece *et al.*, 1998].

At the beginning [Pearl, 1988] BLP was applied directly on graphs even if the structure was not a polytree and has loops. A theoretical problem in this method is that some information can be taken into account two or more times. A more developed BLP technique works directly on Markov Random Fields (MRF), and specially on pairwise MRFs. Like in the junction tree, the potential defined on large sets of nodes increase the computing time since to convert a general MRF to a pairwise one a conglomerate of nodes could be so large.

An option to work with BNs as well as with MRFs is to pass through a Factor Graph [Kschischang *et al.*, 2001]. In such a graph each node send a message to each factor node and each factor node send a different message to each node.

6.5.2 Sampling Methods

The basic idea in all those techniques consists in generating N samples using the joint probability function and the evidence. Then, those samples will be used to compute approximated probability values by using the appearing frequency of events and the total size of samples.

In general, those approximative methods can be classified into two classes. In one hand the stochastic simulation methods and in the other hand deterministic search methods. The way to produce the samples from the joint probability density function is the basic difference between both methods. The firsts produce samples from the joint probability function using some random mechanisms, while the seconds generate the samples in a systematic way.

Basic Simulation Concepts

As it was said before, the basic idea is to produce a number N of samples from a probability density function. Then, uses those samples to compute the frequency of each single event to obtain the searched probabilities. The procedure is not complicated with the exception of the lack of knowledge of the joint probability function. To solve this problem another function, which is easier to simulate, can be used to generate the samples. Those samples must be weighted by another function that measures the similarity between both functions.

Samples will be simulated from the probability density function $p(x)$ which can be written as :

$$\begin{aligned} p(x) &= \frac{p(x)}{h(x)} h(x), \\ &= s(x)\ h(x), \end{aligned} \tag{6.24}$$

where $h(x)$ is the other density function that is easier to simulate and $s(x)$ is a weighting function that measure, the similarity between $p(x)$ and $h(x)$. At the end the weights are normalized to finally compute probability of the samples.

General methodology is as follows. consider a set of random variables $X = \{X_1, \ldots, X_M\}$ with a joint probability density function $p(x)$. Consider, also, the subset E be the evidence with values $\{e_1, \ldots, e_M\}$. The posterior probability of Y a subset of X given $E = e$ is computed from the next equation :

$$p(y|e) = \frac{p_e(y)}{p(e)} \propto p_e(y), \tag{6.25}$$

where

$$p_e(y) = \begin{cases} p(y, e), & \text{if } y \text{ is consistent with } e, \\ 0, & \text{otherwise.} \end{cases} \tag{6.26}$$

The values of $p(y|e)$ are computed from the N samples of $p(x)$ using a different probability density function $h(x)$ as seen before. Then, the probability $p(y|e)$ is approximated with the sum of all the weights consistent with the events y and e. That is, given N samples $x^j = \{x_1^j, \ldots x_M^j\}$ for $j = 1, \ldots, N$, the posterior probability of Y is :

$$p(y|e) \approx \frac{\sum_{y \in x^j} s(x^j)}{\sum_{j=1}^{N} s(x^j)}. \tag{6.27}$$

The reached quality depends on the following factors. First, the function $h(x)$ used to obtain the samples. Second, the method used to generate the samples from $h(x)$. And last, the size of samples N.

A particular case is obtained when both distributions $p(x)$ and $h(x)$ can be written as:

$$p(x) = \prod_{i=1}^{n} p(x_i|\pi_i) \tag{6.28}$$

and

$$h(x) = \prod_{i=1}^{n} h(x_i|\pi_i) \tag{6.29}$$

where π_i is a subset of X. Then, the sample's weight is computed just with the product of the weights as :

$$\begin{aligned} s(x) &= \frac{p(x)}{h(x)} \\ &= \prod_{i=1}^{n} \frac{p(x_i|\pi_i)}{h(x_i|\pi_i)} \\ &= \prod_{i=1}^{n} s(x_i|\pi_i) \end{aligned} \tag{6.30}$$

This simplification is well adapted to BN because they can be expressed in the way (6.28), where as is was already defined, π_i are the parents of x_i.

From the last paragraphs it can be seen that all the methods consist of three components: a distribution $h(x)$ for simulation, a method for generating the samples from $h(x)$, and a formula to compute the weights. Existing methods differ from each other in one or several of those components.

6.5.3 Rejection Sampling (logic sampling)

This method is due to [Henrion, 1988]. The simulation proceeds in a forward way, that is, each variable is generated only if its parents are already sampled. It uses an uninstantiated BN which in each simulation assigns random values to all the variables including also the variables corresponding to observations. If the sample matches the observed data, it is counted, otherwise it is rejected. At the end the belief distributions are calculated by averaging the frequency of counted events. Each X_i is simulated using the associated conditional probability function :

$$h(x_i|\pi_i) = p(x_i|\pi_i), \text{ where } i \in \{1, \ldots, n\}. \tag{6.31}$$

In order to use this method an order for the variables has to be given. The variables are ordered in such a way that the parents always precede their children. This order is called **ancestral order**. Once the parents of X_i have been simulated, that is, they have values assigned, X_i can be simulated using $h(x_i|\pi_i)$ which is in this case $p(x_i|\pi_i)$. Then the weights are computed as :

$$\begin{aligned} s(x) &= \frac{p_e(x)}{h(x)} \\ &= \frac{\prod\limits_{X_i \notin E} p_e(x_i|\pi_i) \prod\limits_{X_i \in E} p_e(x_i|\pi_i)}{\prod\limits_{X_i \notin E} p(x_i|\pi_i) \prod\limits_{X_i \in E} p(x_i|\pi_i)}, \end{aligned}$$

where :

$$s(x) = \begin{cases} 1, & \text{if } x_i = e_i \; \forall \; X_i \in E, \\ 0, & \text{otherwise.} \end{cases} \tag{6.32}$$

If $x_i \neq e_i$ for any $X_i \in E$, then, the weight is zero and the sample is rejected. As it can be seen the draw samples are rejected when they contradict the evidence. Therefore, the method is very inefficient if the evidence, because of a large number of samples, must be rejected and then the process can take more time in order to generate the required N samples.

6.5.4 Likelihood Weighting

This method, due to [Fung and Chang, 1990; Shachter and Peot, 1990], avoids the inefficiency of rejection sampling by generating only events which are consistent with the evidence. Where $h(x_i)$ is in this case :

$$h(x_i) = \begin{cases} p_e(x_i|\pi_i), & \text{if } X_i \notin E, \\ 1, & \text{if } X_i \in E \text{ and } x_i = e_i, \\ 0, & \text{otherwise.} \end{cases} \tag{6.33}$$

Again an ancestral order is needed and the weights for the values taken by X are :

$$\begin{aligned} s(x) &= \frac{p_e(x)}{h(x)} \\ &= \prod_{X_i \notin E} \frac{p_e(x_i|\pi_i)}{p(x_i|\pi_i} \prod_{X_i \in E} \frac{p_e(x_i|\pi_i)}{1} \\ &= \prod_{X_i \in E} p_e(x_i|\pi_i) \\ &= \prod_{X_i \in E} p(e_i|\pi_i), \end{aligned} \tag{6.34}$$

where the last equality is due to that $x_i = e_i$ when $X_i \in E$.

6.5.5 Markov Chain Simulation

Unlike the previous methods, which generate each event independently, Markov chain methods generates each event by making a random change to the preceding one. Therefore, it is important to see the network as being in a particular state which specify a particular value for each variable. The next state is generated by randomly sampling a value for one of the non evidence variables.

The theory of Markov chains is a well developed theory. A Markov chain is a series of random variables, $\{X_1, X_2, \dots\}$, in which the influence of the values of $\{X_1, X_2, \dots, X_n\}$ on the distribution of $\{X_{n+1}\}$ is mediated completely by the value of X_n, that is :

$$P(x_{n+1}|x_1, x_2, \dots, x_n) = P(x_{n+1}|x_n) \tag{6.35}$$

A Markov chain can be specified by giving the marginal distribution for X_1, the initial probabilities of the various states, and the conditional distributions for X_{n+1} given the possible values for X_n called the the transition probabilities for one state to follow another state.

In this method, the next state is generated by sampling one of the non-evidence variables X_i conditioned on the current values of the variables in the Markov blanket of X_i. Monte Carlo Markov Chain (MCMC) technique therefore obtain a sample by randomly search in the space of possible complete assignments, that is, searching in the state space.

Since the sampling process settles into a "dynamic equilibrium" in which the long-run fraction of time spent in each state is exactly proportional to its posterior probability the obtained samples from MCMC are consistent estimates for the posterior probability. This important property is obtained from the specific transition probability with which the process moves from one state to another, as defined by the conditional distribution given the Markov blanket of the variable being sampled.

Let $P(X \rightarrow X')$ be the probability that the process makes a transition from state X to state X'. If the Markov chain is run for t steps, the probability that the system is in the state X is $\pi_t(x)$. Similarly, $\pi_{t+1}(x')$ is the probability of being in state x' at time $t+1$. Given $\pi_t(x)$, the probability $\pi_{t+1}(x')$ can be computed by summing up all the states the system could be in at time t, from the probability of being in that state times the probability of making the transition to x'.

$$\pi_{t+1}(x') = \sum_x \pi_t(x) P(X \rightarrow X'). \tag{6.36}$$

The chain reaches its stationary distribution when $\pi_t = \pi_{t+1}$. The stationary distribution π, is defined in the next equation :

$$\pi(x') = \sum_x \pi(x) P(X \rightarrow X') \ \ \forall\, x'. \tag{6.37}$$

This equation expresses an equilibrium between the inflow and outflow of states. This property, the **detailed balance** property, can be interpreted as an equal flow between any pair of states :

$$\pi(x) P(X \rightarrow X') = \pi(x') P(X' \rightarrow X) \ \ \forall\, x, x'. \tag{6.38}$$

It can be shown that detailed balance property implies stationarity.

To justify this approach it has to be establish that MCMC which defines a transition probability in the sampling step verify the detailed balance property with an stationary distribution equal to $P(X|e)$. To do that, the first step is to use a **Gibbs sampler**. Gibbs sampling starts with a random setting of states and at each step of the sampling process the state variable is update stochastically according to its probability distribution conditioned on all the other state variables. Let X_i be the variable to be sampled with value x_i

in the current state, and $\mathbf{X}_j$ with values $\mathbf{x}_j$, also in the current state, the remainder variables where $i \neq j$. When sampling a new value x_i' conditioned on all the other variables, the transition probability is expressed in the next equation :

$$P(X \rightarrow X') = P((x_i, \mathbf{x}_j) \rightarrow (x_i', \mathbf{x}_j)) = P(x_i'|\mathbf{x}_j, \mathbf{e}). \tag{6.39}$$

It can be proved that this equation, the Gibbs sampler, is in detailed balance with the true posterior probability as follows :

$$\begin{aligned} \pi(\mathbf{x})P(X->X') &= P(\mathbf{x}|\mathbf{e})P(x_i'|\mathbf{x}_j, \mathbf{e}) \\ &= P(x_i, \mathbf{x}_j|\mathbf{e})P(x_i'|\mathbf{x}_j, \mathbf{e}) \\ &= P(x_i|\mathbf{x}_j, \mathbf{e})P(\mathbf{x}_j|\mathbf{e})P(x_i'|\mathbf{x}_j, \mathbf{e}) \\ &= P(x_i|\mathbf{x}_j, \mathbf{e})P(x_i'|\mathbf{x}_j, \mathbf{e}) \\ &= \pi(\mathbf{x}')P(X' \rightarrow X). \end{aligned} \tag{6.40}$$

In a BN, if a variable is independent of all other variables given its Markov Blanket, then the factor $P(x_i'|\mathbf{x}_j, \mathbf{e})$ can be reduced. An example of MCMC applied to BN is showed in [Pearl, 1987b]. It consists in using the known values for the evidential variables and then, simulate the remainders with the probability functions conditioned to the others. A needed initial sample can be generated using one of the previous described methods. Then, the simulation is done for all the other variables without any specific order using the next theorem [Pearl, 1987b] :

For Bayesian Networks the conditional probability function of one variable X_i, conditioned to the remainder variables is given by :

$$h(x_i) = p(x_i|x \setminus x_i) \propto p(x_i|Pa(x_i)) \prod_{X_j \in C_i} p(x_j|Pa(x_j)) \tag{6.41}$$

where C_i is the set of children of X_i and $X \setminus X_i$ the variables of X which are not in X_i.

Once a sample is generated, it is used to produce the next one. It has to be noticed that only X_i, its parents, its children and its children's parents, the **Markov Blanket**, are used to compute $h(X_i)$. Also, the values used for the variables not yet sampled are the previous ones. The Markov methods overcome the rejection problem present in the previous techniques, but has a problem. Some convergence problems appear when extreme probability values are present given that the consecutive samples are not independent.

6.5.6 Maxima Probability Search

This method unlike to the previous ones produces the samples in a deterministic way. The process is based on building a tree. In this tree each branch is associated to a partial event $\{x_1^i, \ldots, x_m^i\}$. At each step a branch is chosen. If $m = n$, that is, if the event is complete, the branch is cut of and included in the final simples. Otherwise, the tree is increased with so many branches as possible values can take the following variable x_{x+1}. Thus, the original branch $\{x_1^i, \ldots, x_m^i\}$ is replaced by the branches $\{x_1^i, \ldots, x_m^i, x_{m+1}\}$ for all the possible values of X_{m+1}.

Some techniques which use this process have been proposed along the last decades. For example, [Pearl, 1987a; Henrion, 1991; Poole, 1993; Srinivas and Nayak, 1996]. All those techniques differ just in the criterion used to choose the branches. One option is that used by [Poole, 1993] where the maxima probability is the measure. The algorithm presented is an example of those techniques.

Given an ancestral order in the variables $X = \{X_1, \ldots, X_n\}$, the tree is started with so many branches as possible values can take the first variable X_1. Then, the probability for each branch is computed, and the one with maximum probability is chosen. In the next step the selected branch is augmented with a sub-tree

formed by the branches associated to the possible values of the second variable. This process continues until finish with the last variable X_n. At this moment the branch with maximum probability corresponds to the first sample. Using the described process, after couped the used branch, the second sample is generated using the branch with the maxima probability among the unremoved branches.

6.6 Conclusions

This chapter was intended to provide with the necessary background about inference which can help the reader to understand the process used into the remainder of this work. Principles and techniques for exact as well as approximate inference were described in this chapter. Inference is a very important process in BNs. It allows to compute the probability of hidden variables once some of the other have been instantiated. From its definition it can be seen that inference is an essential procedure for parameter estimation techniques which will be used in our SV system based on BNs.

Next chapter is dedicated to learning. The possible source of information obtained from the speech will be related to each other using a graphical model. The structure of this model will be deduced using techniques of structure learning which will be described in the next chapter. Parameters estimation will be also addressed.

Chapter 7

Learning in Bayesian Networks

After inference, learning is the second main problem in probabilistic reasoning with BN [Heckerman *et al.*, 1995; Buntine, 1996; Fisher and Lenz, 1996; Castillo *et al.*, 1997]. In this chapter the two components of BNs, the structure and parameters learning is addressed. Learning BN from data consists in automatically obtaining the structure and parameters from information in the available samples. It will be seen that structure learning is a much more difficult task. In general, there are four recognized variants of this problem. They are related to the knowledge of the structure and also to the database. The used database to learn the BN can be full observed or just partially observed.

Statistics is the basis on which BNs are founded for the development of learning algorithms. In the first section structure learning techniques will be presented. These techniques are used to obtain the conditional independencies in the graph directly from databases. Algorithms for learning BN from data can be grouped into two categories. Network scoring and conditional independence test algorithms. The first one computes a score that reflects the degree of match between the given structure and the data. The second looks into the relationships between variables to build the structure.

In a second section we will address the parameters learning problem. Those parameters are the conditional probability distributions that quantify the conditional dependences present in the graph. This problem will be continue in the next chapter where the adaptation technique will be described.

7.1 Structure Learning

One way to obtain the probabilistic relationships among the variables is asking an expert who can gives a possible good structure. However, that structure may not reflect all conditional independencies present in the data. Another way to obtain the structure is with the chain rule, a set of conditional independencies and any ordering on the variables. But, maybe the easiest way, or not, to proceed is to score all the possible structures and takes the best one. The number of possible structures depends on the number n of variables in a super-exponential way. Then, it is unrealistic to explore all of them for a high value of n. In [Robinson, 1977] has been proved that the number of possible DAGs (section 4.2.2) $G(n)$ with n variables is given by the next equation:

$$G(n) = \sum_{i=1}^{n} (-1)^{i-1} \left(\begin{array}{c} n \\ i \end{array} \right) 2^{i(n-i)} G(n-1), \tag{7.1}$$

where $G(0) = 1$. Using this equation the values for a given n can be obtained. For example $G(2) = 3$, $G(3) = 25$, $G(4) = 543$, $G(5) = 29281$ and so on. However, it happens that several of these are equivalent since they represent equivalent independence statements, but is not the case if the causality is taken into account. For example for $n = 3$, $G(3) = 25$ but there are just 11 different equivalence classes of networks if the causality is absent.

7.1.1 Structure Search

Independence or/and dependence test, like PC algorithm [Spirtes *et al.*, 1993], and search and scoring algorithms, like K2 [Cooper and Herskovits, 1992], are the two main approach for Structure learning. Most of actual structure learning algorithms belong to the last approach which have two components. The first one search for the structure in the full or limited structure space and the second one score the structure. In the first step some constraints could be imposed to the structure space in order to reduce the possibilities, or to impose some known restrictions. There are basically two options, trees as a basic structure or general complex structures.

There are two different approaches to find the best structure. The first one, like MCMC, searches in all the structure space and returns either the best one, or the best in a Markov equivalent way. Alternatively, it can be started with a specific connected graph and then searches for independence relations in the data, and puts in, takes away or reversing arcs to make modifications, re-tuning the parameters after each changing in the structure. Given that the cycles are forbidden some algorithms assumes an order in the variables, an ancestral order. Therefore, a node can have parents just among the nodes that comes earlier in this order.

The second component in search and scoring algorithms is the scoring. A first approach is to use the likelihood function, but it privileges the fully connected graphs. Adding more parents to a node, and more parameters to the model, cannot decrease the likelihood. To overcome this problem a penalizing factor is added to the scoring function. This extra factor is intended to penalize complex structures. In general the score function are decomposable. That is, the score is the product or the sum of the score of families in the structure (node and its parents). This property let to re-compute just the contribution to the full score that comes from the changed families. MAP or Minimum Description Length (MDL) approaches subtract a penalty factor from the likelihood before comparing different structures. The Bayesian approach places a joint prior over structures and parameters.

7.1.2 PC Algorithm

This algorithm belongs to the independence or/and dependence test class. The basic idea is to measure the conditional independence between a pair of variables given a set of other variables. Then, those obtained conditional independence are used to build the structure. In practice a full connected graph is used as initialization. After an independence is revealed the concerned arc is removed.

To test the conditional independence in the previous cited conditions the cross entropy (CE) can be computed. The next equation define the CE for X and Y given Z:

$$CE(X,Y|Z) = \sum_{Z} P(z) \sum_{x,y} P(x,y|z) \ \log\left(\frac{P(x,y|z)}{P(x|z)\ P(y|z)}\right), \tag{7.2}$$

where the probabilities are the maximum likelihood computed from the database.

As all algorithm, the obtained results depend on the quality of the database. All the conditional independencies should be present in the data. The tested variables can be conform to some prior knowledge. For example, an ancestral ordering, the existence of one or several edges or to some edge orientation.

7.1.3 Greedy Search Algorithm

This algorithm comes from the optimization area and it is also know as hill climbing. It works with all possible structures. It starts with a given structure, where some knowledge about the relationship between the variables is presented. From this first structure a set of them close to the first are defined. Those close structures are the ones which are obtained putting some new edges, taking away or reversing some other

edges in the original one.

Once the set of close structures is generated an score is assigned to each one. The algorithm is finish when a maximum is reached, that is, when changes to the actual structure does not increase the score.

7.1.4 K2 Algorithm

K2 algorithm [Cooper and Herskovits, 1992] belongs to the second class of methods. It starts with a structure, the simplest one, i.e. a graph without arcs. The searched structure space is restricted to an small subset using some prior knowledge. This restriction is expressed in a relationship between the variables, for example an ancestral order. Then, for each variable X_i the set $Pa(X_i)$ is searched. The variables in this set ($Pa(X_i)$) are restricted to those variables with smaller order numbers than X_i which still too large. Therefore, instead of using all the possible variables, it is suggested to use a greedy heuristic algorithm assuming that a node in the structure has no parents and incrementally adding parents with the preceding restriction whose addition most increases the probability of the structure until no improvement is obtained, a threshold or a maximum number of parents is reached.

Bayesian approach used in this work assumes a uniform prior over all possible network structures. Let S be a data base of samples with m cases corresponding to instantiations of a set of variables $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$. If G is the structure, the problem is to compute the probability of that structure given the data, that is, $P(G|S)$. To solve this problem some assumptions are done. The first one is the nature of variables, they are supposed to be discrete. This first assumption let to write the next joint probability:

$$P(G, S) = \int_{G_p} P(S|G, G_p) f(G_p|G) dG_p,$$

where G_p is a conditional probability assignment over all variables and f is a conditional density function.

The second assumption establishes that cases occur independently. Then the conditional probability function of data given the structure and conditional probability assignments is :

$$P(S|G, G_p) = \prod_{i=1}^{m} P(C_i|G, G_p),$$

where C_i represent the cases and $P(C_i|G, G_p)$ is computed directly from the structure and the conditional probability assignments.

The distribution of$(G_p|G)$, even if other possibilities are mentioned, is assumed to be uniform.Finally, the last assumption is that variables are fully observed. This last constraint can be overcome if some missing data are present. A possibility is a standard method such as Gibbs sampling for dealing with those missing values or just a inference technique.

Before pursuing let's redefine some conventions. Each variable X_i has an associated collection of values $\{x_i(1), \ldots, x_i(p)\}$ that it can take on, where the number of value p depends on i. A set of variables $\mathbf{X}$ which take the values $\{X\_1 = x_1, X_2 = x_2, \ldots, X_N = x_N\}$ will be written as $\mathbf{X} = \mathbf{x}$. Given X_i its parents are $Pa(X_i)$, k is the instantiation for the parents.

Define N_{ijk} to be the cases in S in which variable $X_i = j$ and the parents $P(X_i)$ are instantiated as k, then:

$$N_{ij} = \sum_{k=1}^{p} N_{ijk}.$$

Given the four assumptions [Cooper and Herskovits, 1992] shows that the joint probability of data and graph can be written as :

$$P(G,S) = P(G)\prod_{i=1}^{N}\prod_{j=1}^{q_i}\frac{(p_i-1)!}{(N_{ij}+p_i-1)!}\prod_{k=1}^{p_i}N_{ij}! \tag{7.3}$$

Finally, the conditional probability of the structure given the data is :

$$\begin{aligned} P(G|S) &= \frac{P(G,S)}{P(S)} \\ &= \frac{P(G,S)}{\sum_{G\in Q}P(G,S)}, \end{aligned}$$

where Q is the possible family of structures. This algorithm permits to compare just two different structures. It is impractical to search over all the possible structures or compute $max_G P(G|S)$ if N starts to be an important number. To solve this problem it is proposed to restrict the research to a small subset of Q as it was established in the last section.

7.1.5 Bayesian Information Criterion

By Bayes' rule, the MAP model is the one that maximizes an structure G given the data S :

$$P(G|S) = \frac{P(S|G)P(G)}{P(S)}, \tag{7.4}$$

where $P(G)$ penalizes complex model and $P(S)$ is a constant. The marginal likelihood is :

$$P(S|G) = \int_\theta P(S|G,\theta)P(\theta|G)d\theta. \tag{7.5}$$

This equation has the advantage to automatically penalizes more complex structures. This score function can be approximated [Heckerman, 1998] with a Laplace method, and finally it is obtained the BIC (Bayesian Information Criterion) :

$$\log P(S|G) \approx \log P(S|G,\hat{\theta}) - \frac{d}{2}\log M, \tag{7.6}$$

where M is the number of samples, $\hat{\theta}$ is the ML estimate of the parameters and d is the dimension of the model.

7.1.6 MDL Approach

A disadvantage of using an uniform prior is that the chosen model could be a complex one. Even if is last model is just slightly better than another one less complex [Lam and Bacchus, 1994]. From empirical and some theoretical results it is known that less complex models are often more accurate. The Minimum Description Length (MDL) approach privilege simple model.

The basic idea in MDL is that the best model for a database is that one which minimizes the sum of two term, both measured in bits. The first term is used to encode the length of the encoding of the model and the second the length of the encoding of the data given the model. In this context it is useful to think about a model as a mean of compressing data to any desired accuracy.

To encode a BN with n nodes it is needed just to encode a list of the parents of each node and a set of conditional probabilities for each node because this first list gives already a description of the nodes in the network. For a node with k_i parents it is needed $k_i \log n$ bits to encode the list of its parents. The encoding, the size of the conditional probability table, for a given node depends on the number of parents and the number of possible values that the variables can take on. The total description length of the BN requires the next quantity in bits :

$$\sum_{i=1}^{n} k_i \log n + d(s_i - 1) \prod_{j \in F_i} s_j, \tag{7.7}$$

where d is the number of bits required to encode a probability, F_i is the set of parents of node i and $s - 1$ is due to the fact that probabilities of the instantiations of any variable sums to one.

Given the frequency of occurrence, or probability, of samples in the data a character code which gives shorter codes to frequently occurring samples can be used to encode the data using the model. Given those frequency the Huffman's algorithm provides a method for generating optimal character codes. The probabilities in a BN can be used to generate a Huffman code and then, usingthis code to measure the encoding length of the data given the model. Let $P = \{p_1, \ldots p_m\}$ a distribution over m possible events $\{e_1, \ldots e_m\}$, where p_i is the probability of e_i. Huffman's algorithm gives to each e_i a codeword of length approximately equal to $-logp_i$. For a sequence of N events it can be expected to have N p_i occurrences of e_i and then the encoding of the sequence is :

$$-N \sum_{i=1}^{m} p_i \ \log \ p_i,$$

which is the optimal encoding. If the true values for P are unknown some approached values $Q = \{q_1, \ldots, q_m\}$ can be used :

$$-N \sum_{i=1}^{m} p_i \ \log \ q_i. \tag{7.8}$$

In general, this quantity can be difficult to compute because of the number of events which should be large. Instead, given that the encoding length of the data is a monotonically increasing function the cross entropy between the distribution defined by the model and the true distribution is used. Let the Kullback Leibler function be :

$$C(P, Q) = \sum_{i=1}^{m} p_i \log \left(\frac{p_i}{q_i} \right).$$

Until now, the problem is stil there, but using a decomposition of the events in terms of the BN structure $P(X_i|Pa(X_i))$, the function $C(P, Q)$ is a monotonically decreasing function of :

$$\sum_{i=1:Pa(X_i)\neq\emptyset} W(X_i, Pa(X_i)),$$

where

$$W(X_i, Pa(X_i)) = \sum_{X_i, Pa(X_i)} P(X_i, Pa(X_i)) \ \log \left(\frac{P(X_i, Pa(X_i))}{P(X_i)P(Pa(X_i))} \right).$$

All the probabilities are computed from the data and then, this equation is used to compute the encoding length of the data instead of equation (7.8).

7.1.7 Tree-structures using a MDL Approach

A different approach to learn BN tree-structures using a MDL approach is described in [Sigelle, 2003]

From equation :

$$P(S|G) = \int P(S|G,\theta)P(\theta|G)d\theta, \tag{7.9}$$

and from its limited expansion up to the second order one has the next equation:

$$P(S|G) \approx L(\hat{\theta}) \int exp - \frac{1}{2}(\theta - \hat{\theta})^t \, \mathbf{A}(\theta - \hat{\theta})d\theta \approx L(\hat{\theta})(\frac{1}{2\pi})^{\frac{d(G)}{2}} \frac{1}{\sqrt{det\ \mathbf{A}}}, \tag{7.10}$$

where $d(G)$ is the number of parameters specifying the model G. At the lowest order if N is the number of observations the MDL equation is obtained :

$$\mathcal{L} = \log P(S|G) \approx L(\hat{\theta}) - \frac{d(G)}{2} \log N. \tag{7.11}$$

Now, for a tree structure and in particular when all the nodes are independent it can be written :

$$\log P(S|G) \approx \sum_{s \in G} \{[\sum_{i \in \Omega^s} (N_i^s) + \frac{1}{2} \log N_i^s] - (N + |\Omega^s| - \frac{1}{2}) \log N + (|\Omega^s| - 1) \log \sqrt{2\pi}\}, \tag{7.12}$$

where Ω^s is the set of observable states at generic node s, and N_i^s is the observed number times of the variable s is in the state i.

7.2 Parameters Learning

Once the structure is established the parameters are the only component to be computed in order to have a complete BN. It is required to adjust the parameters of the BN in such a way that the CPDs describe the data statistically. As before, the characteristics of the database are important.

7.2.1 Known Structure and Full Observability

Given that all the variables are observed, that is, completely observed the problem can be decomposed into a series of terms. Given a set of samples $x = \{x(1), \ldots, x(t), \ldots, x(T)\}$ which have been draw independently from a probability law $p(x|\theta)$ the Maximum Likelihood approach can be used.

For the case of a multinormal distribution the parameters are written in the following form:

$$\theta_{ijk} = P(x_i = j|Pa(x_i) = k), \tag{7.13}$$

where θ_{ijk}, the actual parameter, represent the probability that the variable x_i is in the state j and its parents $Pa(x_i)$ in the state configuration k.

For a multinomial distribution the sufficient statistics are just counting of possible configurations in each CPT. Then, if M samples are observed, this counting is computed as is represented in the next equation:

$$N_{ijk} = \sum_{t=1}^{T} \mathbf{1}_{(x_i(t)=j, Pa(x_i(t))=k)}.$$

Then, the log likelihood can be written as follows:

$$\begin{aligned} \mathcal{L} &= \log \prod_t \prod_{ijk} \theta_{ijk}^{N_{ijk}}, \\ &= \sum_t \sum_{ijk} N_{ijk} \log \theta_{ijk}. \end{aligned}$$

To optimize this equation the Lagrange multiplier enforced to $\sum_j \theta_{ijk} = 1$ should be used to finally obtain the optimal value :

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{\sum_{j'} \theta_{ij'k}}. \tag{7.14}$$

If this law $p(x|\theta)$ follows a normal distribution $\mathcal{N}(\mu_i, \Sigma_i)$ the optimal values $\{\hat{\mu}, \hat{\Sigma}\}$ for a multivariate normal distribution are [Duda and Hart, 1973] :

$$\begin{aligned} \hat{\mu} &= \frac{1}{T}\sum_{k=1}^{T} x(k), \\ \hat{\Sigma} &= \frac{1}{T}\sum_{k=1}^{T} (x(k) - \hat{\mu})(x(k) - \hat{\mu})^t. \end{aligned}$$

The other parameters to be computed are now the relation of dependence between two normal variables. This relation is expressed in a multivariate linear regression expression. The relationship between two such variables is represented on the following equation :

$$p(y|x) = (2\pi)^{d/2}|\Sigma|^{-1/2} \exp[-\frac{1}{2}(y - Ax)^t \Sigma^{-1}(y - Ax)] \tag{7.15}$$

for $y \in R^d$ and $x \in R^k$, where the regression matrix $\mathbf{A} \in R^{k \times d}$ and $\mathbf{r}$ is the precision matrix, the inverse of the covariance matrix Σ.

To optimize the value of A under the ML technique the gradient is used. Then, if the log likelihood is expressed as follows :

$$\mathcal{L} = -\frac{1}{2}\sum_t \log|\Sigma| - \frac{1}{2}\sum_t (y(t) - Ax(t))^t \Sigma^{-1}(y(t) - Ax(t)),$$

its derivative is obtained using the following expression :

$$\frac{\partial}{\partial \mathbf{M}}(\mathbf{M}a + b)^t \mathbf{C}(\mathbf{M}a + b) = (\mathbf{C} + \mathbf{C}^t)(\mathbf{M}a + b)a^t,$$

the derivative is :

$$\begin{aligned} \frac{\partial}{\partial \mathbf{A}} &= -\frac{1}{2}\sum_t 2\Sigma^{-1}(y(t) - Ax(t))x(t)^t \\ &= -\Sigma^{-1}\sum_t y(t)x(t)^t - A\sum_t x(t)x(t)^t. \end{aligned}$$

An finally the optimal value for A is :

$$\hat{A} = -\Sigma^{-1}\left(\sum_t y(t)x(t)^t\right)\left(\sum_t x(t)x(t)^t\right)^{-1}. \tag{7.16}$$

This matrix depends only on the covariance between the variables.

7.2.2 Known Structure and Partial Observability

Unlike the fully observed case, the partial one can not be decomposed into a product of local terms because of the hidden or missing variables. To overcome this problem the most employed techniques is the Expectation Maximization (EM) algorithm ([Dempster *et al.*, 1997] in general or [Lauritzen, 1995] for graphical

models).

The idea behind EM is to suppose first the possible parameters of the model, or BN in our case, and then compute the probability for each samples. Using these probabilities, the parameters can be computed in a second step. The first step is called Expectation, which computed the expected values of the log likelihood of the completed data with respect to the posteriori over the hidden variables. The second step, called Maximization, is where expected log likelihood is maximized with respect to the parameters to obtain more refined values. Those two steps are repeated until convergence.

In the maximum likelihood estimation problem a density function $p(\mathbf{x}|\Theta)$ depends on a set of parameters θ which are unknown, for example the means, weight and variance for a GMM. The second element in this problem is a sequence of observation of size N which are supposed to be drawn from that distribution, $\mathbf{X} = \{X_1, \ldots, X_N\}$. If those data vectors are i.i.d. with a distribution p, therefore the density, or likelihood of the parameters given the data, for the samples can be written as :

$$p(\mathbf{X}|\Theta) = \prod_{i=1}^{N} p(x_i|\Theta), \tag{7.17}$$

The likelihood is a function of the parameters Θ where the data are fixed. Now, for our problem, the maximum likelihood computation, the parameters Θ which maximize that function are searched :

$$\hat{\Theta} = arg\max_{\Theta} p(\mathbf{X}|\Theta). \tag{7.18}$$

For minimization complexity reasons usually the log likelihood is computed.

If $\mathbf{X}$ are the incomplete data and it is assumed that a complete data set is $\mathbf{Z} = \{\mathbf{X}, \mathbf{Y}\}$ with the following joint density function :

$$p(\mathbf{z}|\Theta) = p(\mathbf{x}, \mathbf{y}|\Theta) = p(\mathbf{y}|\mathbf{x}, \Theta)\ p(\mathbf{x}|\Theta). \tag{7.19}$$

With this new density function the complete data likelihood can be defined as the probability $p(\mathbf{X}, \mathbf{Y}|\Theta)$. Given that $\mathbf{Y}$ is random, unknown and follows an underlying distribution it can be treated as a random variable.

The first step for the EM algorithm, the Expectation step, is to find the expected values of the log likelihood of the complete data with respect to the unknown data $\mathbf{Y}$ given the observed data $\mathbf{X}$ and the parameters Θ in the following way :

$$Q(\Theta, \Theta^{n-1}) = E[log\ p(\mathbf{X}, \mathbf{Y}|\Theta)|\mathbf{X}, \Theta^{n-1}], \tag{7.20}$$

where Θ^{i-1} are the current parameters and Θ are the new parameters. It should be noticed that $\mathbf{X}$ and Θ^{i-1} are both constants, Θ is the variable to be adjusted, and $\mathbf{Y}$ is a random variable which follows a distribution written here as $f(\mathbf{y}|\mathbf{X}, \Theta^{i-1})$. Using this definition the expected value can be written as is shown in the next equation :

$$E[\log\ p(\mathbf{X}, \mathbf{Y}|\Theta)|\mathbf{X}, \Theta^{n-1}] = \int_{\mathbf{y}\in\Upsilon} \log\ p(\mathbf{X}, \mathbf{Y}|\Theta) f(\mathbf{y}|\mathbf{X}, \Theta^{n-1}) d\mathbf{y}, \tag{7.21}$$

where Υ is the space of values that $\mathbf{y}$ can take on. From this equation it can be noticed that the results is a constant function that depends just on Θ because the integral is over all possible values of $\mathbf{y}$ and X and Θ^{i-1} are both constants.

The second step, the M step, is the maximization of the expected value issued of the first step, as is show in the next equation :

$$\Theta^n = arg\max_{\Theta} Q(\Theta, \Theta^{n-1}). \tag{7.22}$$

The EM algorithm is an iterative algorithm, that is, both steps, the expectation and maximization are repeated until a convergence criterion as reached. Each iteration guarantes that the likelihood will be increased to converge to a local maximum.

One of the most popular modified version of this algorithm, the generalized EM, computes just a value which increase $Q(\Theta^i, \Theta^{i-1})$ instead of maximize it using a gradient ascent algorithm.

The EM algorithm for a multinomial distribution [Murphy, 2002] defines the expected complete data log-likelihood as follows:

$$Q(\theta^n_{ijk}, \theta^{n-1}_{ijk}) = \sum_{ijk} E[N_{ijk}] \ \log \ \theta^n_{ijk}, \tag{7.23}$$

where $E[N_{ijk}]$ is equal to $\sum_m P(x_i = j, P(x_i) = k|S, \theta)$. Where the probability $P(x_i = j, P(x_i) = k|S, \theta)$ is needed to solve the problem. This step is does in a general inference algorithm. If it is remembered in the junction tree algorithm a tree of cliques is build. Each clique is composed of at least a family of variables, that is, the variable and its parents. Then, doing inference in the junction tree the probability of a family given the data S and the given parameters θ is obtained. Then, the maximization step is defined in the following equation:

$$\hat{\theta}_{ijk} = arg\max_{\theta^n_{ijk}} Q(\theta^n_{ijk}, \theta^{n-1}_{ijk}), \tag{7.24}$$

and the solution is the known equation:

$$\hat{\theta}_{ijk} = \frac{E[N_{ijk}]}{\sum_{j'} E[N_{ij'k}]}. \tag{7.25}$$

7.3 Using Bayesian Networks in Speaker Verification

In this section it is proposed [Sánchez-Soto *et al.*, 2003; 2004a] to integrate, or combine the information (cepstral from the signal $SLPCC$, cepstral from the residual $RMFCC$, pitch F_0 and energy E) at a base level. This information will be combined in a probabilistic framework with a system based on Bayesian Networks (BNs). BNs allows the representation of the conditional independence relations among the proposed variables which convey information about the speaker identity.

As we know, a BN (chapter 5) is a couple $(G, CPDs)$ formed by one structure, the graph G, which is a Directed Acyclic Graph (DAG) and a set of Conditional Probability Distributions (CPD). In order to combine the proposed variables a first step is to find the respective conditional independence relations.

Some of the techniques described in the previously sections will be employed with the development database to search for the structure directly from the data, and in a second step those structures will be used with all the data to compute the parameters of models based on those structures.

7.3.1 Structure Searching using Greedy Search and BIC

The structure search is started using the four vectors ($SLPCC$, $RLPCC$, F_0 and E), and a greedy search algorithm already described (section 7.1) using an *a prior* given by an order into the variables. The used data comes from the database already described in section 3.6. The BN's structures were learned using only the data employed to learn the world model. The search has been performed with all the possible orders and the BIC score like the quality measure [Heckerman, 1998]. From this analysis a probabilistic network structure which has a high posterior probability given the database is obtained. The resulting structure, (Figure 7.1), which is set to be speaker independent, gives the conditional independence relations for the selected variables. This model will be called from now **Model K2-I** because the process of structure search is based on the K2 algorithm (section 7.1.1).

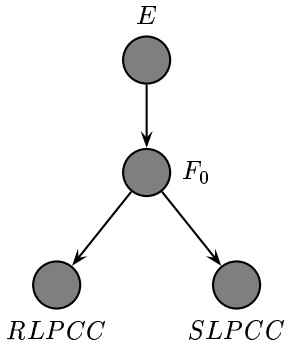


Figure 7.1: *Model K2-I, Structure for the four variables (energy (E), pitch (F_0), signal $SLPCC$ and residual $RLPCC$) issued from the greedy search algorithm with BIC quality measure.*

From basic probability theory the joint probability for the four given variables

$$U = \{SLPCC, RLPCC, F_0, E\}$$

can be written as follows:

$$P(U) = P(E)\ P(F_0|E)\ P(RLPCC|F_0, E)\ P(SLPCC|F_0, E, RLPCC). \tag{7.26}$$

Now, taking into account the graph shows in Figure 7.1 and its relations of conditional independence, the previous equation can be written as a product of local terms. One term associated to each variable in the network:

$$P(U) = P(E)\ P(F_0|E)\ P(RLPCC|F_0)\ P(SLPCC|F_0). \tag{7.27}$$

7.3.2 Physical Interpretation

The conditional independence relations found between the four variables can have an interpretation from the physical point of view. The relation between $SLPCC$, $RLPCC$ and F_0 is obtained from the two last terms in 7.27:

$$P(RMFCC|F_0)\ P(SLPCC|F_0). \tag{7.28}$$

These terms can be interpreted as a relation of conditional independence where $RLPCC$ and $SLPCC$ are independent given the pitch F_0, $RLPCC \perp SLPCC|F_0$. From the second term in 7.27, F_0 depends directly of E given this structure.

The physical interpretation of the relations between the variables gives the same relations found in the equations obtained from the graph. For example, the voiced speech has more energy that the unvoiced speech. It is evident that there exists a close relation between the speech energy and voicing of speech. This fact is written in the term $P(F_0|E)$ in the equation 7.27. The source influences the spectral envelope due to the filtering effect of the vocal tract. The pitch is correlated with the vibration of the vocal folds and the vocal tract characteristics. Consequently, the source and the spectral envelope depend on pitch as it is seen in the next two terms $P(RMFCC|F_0)\ P(SLPCC|F_0)$ in the equation 7.27. The relations obtained in equation 7.27 exhibit a causal, or production interaction between the variables.

7.3.3 Equivalent Model

From a mathematical point of view it can be obtained an equivalent model. Using Bayes theorem : $P(E)\ P(F_0|E) = P(E)\ P(F_0|E)$, the equation (7.27) can be rewritten as follows:

$$P(U) = P(F_0)P(E|F_0)P(RLPCC|F_0)P(SLPCC|F_0). \tag{7.29}$$

This new formulation corresponds to the graph shown on Figure 7.2. This model will be called from now **Model K2-Ib**.

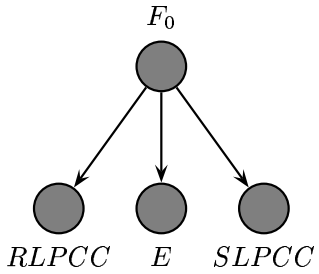


Figure 7.2: *Model **K2-Ib**, Equivalent structure for the four variables (energy (E), pitch(F_0), signal $SLPCC$ and residual $RLPCC$) using the equality $P(E)P(F_0|E) = P(E)P(F_0|E)$.*

From equation 7.29 the causal relations represented are not similar to that presented in 7.27, but the probability density function is the same. Then the equation (7.29) also represents the variables relation present in the joint density. This structure has the advantage that pitch is the root node. Pitch is a feature whose domain is longer than just one single phonetic segment. Then the independence relations found in the equation (7.29) represent the conditional independence of $SLPCC$, $RLPCC$ and E given F_0. Recall that F_0 is a prosodic variable that takes into account different linguistic elements, by making boundaries and defining transitions in speech signal as already seen.

7.4 Structure Searching using MDL

Although the number is structures in a given problem is a rapidly increasing function of the number of variables (7.1) until now only one structure has been obtained. For example, in our problem with four variables, there are 543 possible structures. Therefore, it is possible to find a new structure that could better reflects the conditional independencies presented in the data. In this section two new tree structures are proposed to model the relationships between the four discretized variables using The Minimum Description Length (MDL) approach.

MDL [Sigelle, 2003; Chickering and Heckerman, 1997; Cooper and Herskovits, 1992] (section 7.1.7) analysis was performed using discretized data in order to simplify the probabilistic scheme. Those data were obtained using a Vector Quantization (VQ) technique over all variables in the development database. The initialization was done using a k-means algorithm. $SLPCC$ and $RLPCC$ variables were discretized using 32 values, E with two values and F_0 with three values (one value corresponding to the unvoiced part).

The first obtained structure (Model **MDL-I**) is shown in Figure 7.3. The conditional probability density for the four variables issue of the structure is:

$$P(U) = P(SLPCC)\ P(RLPCC|SLPCC)\ P(F_0|SLPCC)\ P(E|SLPCC). \tag{7.30}$$

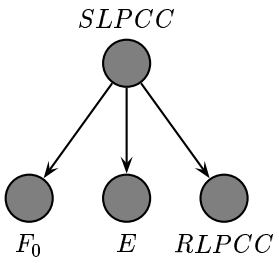


Figure 7.3: *Model **MDL-I**, First structure issued of the MDL analysis.*

7.4.1 Physical Interpretation

Thinking about the $SLPCC$ coefficients computation it is easy to see that those coefficients contain a lot of information. This quantity depends on the number of coefficients (p) used in the autocorrelation function computation. In $SLPCC$ coefficients it can be found the excitation and then also the pitch characteristics since the LP model is not perfect.

An important fact to be remarked in this structure is the the relation between the energy and the cepstral coefficients of the speech signal. The speech cepstral coefficients do not contains the first coefficient, the energy. Then, normally such a relation between the energy and the cepstral coefficients, like it is represented in the structure, should not be exist. But, it has to be remembered that a vector quantification was made in order to obtain the structure. Once the samples have been quantified the obtained index corresponds to some regions of the acoustical space. Therefore the relation founded is correct. This explanation can be applied to the other relation presented in the structure. The obtained relation are between regions of the acoustical space.

7.4.2 A Second Structure

The second structure (Model **MDL-II**) shows in figure 7.4 is equivalent to the first one from the mathematical point of view. The only difference is the causal relationship between the energy E and the $SLPCC$ coefficients. The conditional probability density for the four variables given this structure is:

$$P(U) = P(E)\ P(SLPCC|E)\ P(F_0|SLPCC)\ P(RLPCC|SLPCC), \tag{7.31}$$

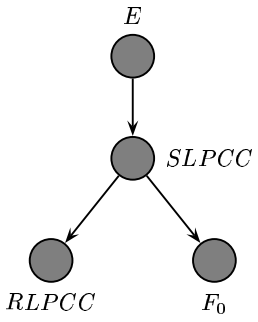


Figure 7.4: *Model **MDL-II**, Second structure issued of the MDL analysis.*

7.5 Structures with GMMs

The relationship between the variables found in the previous section gives the conditional independencies present in the structure. A good and robust way to represent each continuous variable is using GMMs. Then, if all the variables in the model **Model K2-I** depicted on Figure7.1 are represented with GMM (section 5.5) the model can be depicted as done in Figure 7.5, where $\{E, F_0, SLPCC, RMFCC\}$ represent discrete variables and $\{e, f_0, slpcc, rmfcc\}$ continuous variables.

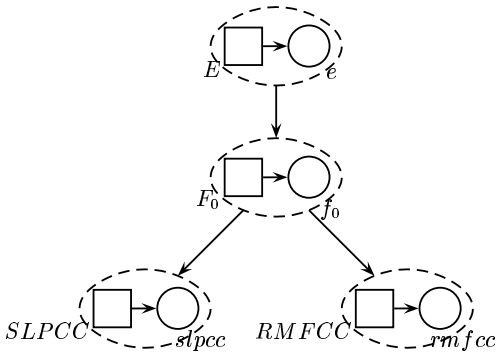


Figure 7.5: ***Model K2-I*** *representation using GMMs for each variable.*

From this last figure all the possible edges which can connect the variables in the model are evident.

Three possibilities occur:

1) **First**, the discrete variables which determine the used gaussian in each GMM can be joined with an edge. That representation would express a relation between the discrete variables which define the gaussian used to represent each variable, (Figure 7.6).

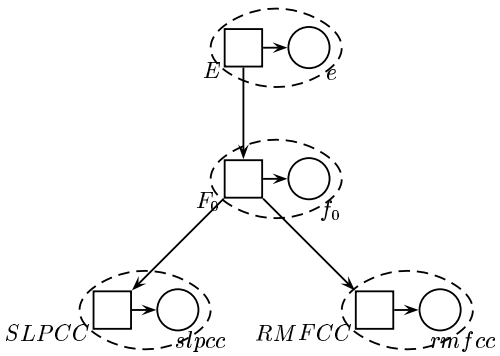


Figure 7.6: ***Model K2-I*** *with relations between the discrete variables. These variables determine the gaussian used in each GMM.*

The associated joint probability density for the eight variables is as follows :

$$\begin{aligned} P(U) &= P(E)\,P(e|E)\,P(F_0|E)\,P(f_0|F_0)\,P(SLPCC|F_0) \\ &\quad P(slpcc|SLPCC)\,P(RMFCC|F_0)\,P(rmfcc|RMFCC). \end{aligned} \tag{7.32}$$

2) **Second**, if the edges are put between the continuous variables, Figure 7.7, the relationships are set between the observed values and not between the discrete variables which define the used gaussians. Then, a direct relationship between the observations is done.

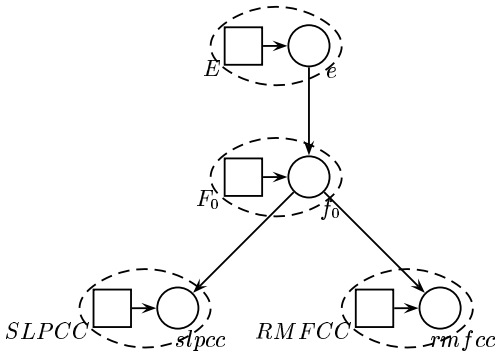


Figure 7.7: ***Model K2-I*** *with a relation between the continuous variables.*

The joint probability density for the case where the edges are between the continuous variables can be written as follows:

$$\begin{aligned} P(U) &= P(E)\,P(e|E)\,P(F_0)\,P(f_0|F_0,e)\,P(SLPCC) \\ &\quad P(slpcc|SLPCC,f_0)\,P(RMFCC)\,P(rmfcc|RMFCC,f_0). \end{aligned} \tag{7.33}$$

3) **A third** option could be to join both, the discrete and continuous variables, but this option has not been studied because of the complexity and time consuming task.

7.6 Applications and Results

In this section the proposed procedure based on BNs to model the available information about the speaker is used. The three structures (**K2-I**,**MDL-I**, **MDL-II**) are tested in the same conditions (same data, same initialization in the context of NIST 2004 speaker recognition evaluation [NIST's 2004 Speaker Recognition Evaluation, 2004] for more details about our work in relation to this NIST's evaluation see Appendix A). Results show the structure influence in the final score. In this part the relation between the continuous variables is used.

Once the structures have been established, the parameters for the final Universal Background Model (UBM) are learned. This model keeps all the information about the independence relations between the variables. As well as for the GMM systems presented in previous section an adaptation of means was performed. In this case a linear combination of mean values of the UBM m_m and speaker model m_s after each EM step is used in all the model from now:

$$m'_s = \nu m_m + (1-\nu)m_s, \tag{7.34}$$

where ν is equal to 0.75 for all the experiments.

7.6.1 Using The Continuous Relations

This set of experiments use the four models obtained in the preceding section, Models **K2-I**, **K2-Ib**, **MDL-I** and **MDL2**. Gaussian Mixtures (GM) were used to represent each variable. Then, for the model **K2-I** the structure is presented in Figure 7.7. This structure is represented again in Figure 7.8 and will be called **Model K2-I-c**, where c is for continuous. The Models **K2-Ib** (**to do**), **MDL-I**, **MDL2** using GM for representing each variable are depicted in Figures 7.9, 7.10 and will be called **K2-Ib-c**, **MDL-I-c**, **MDL–c** respectively.

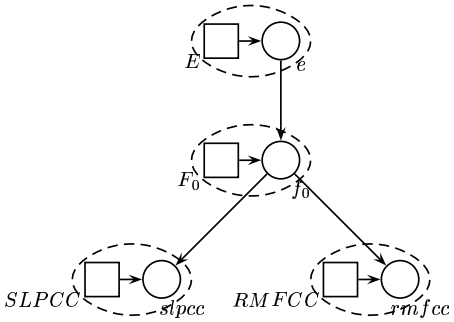


Figure 7.8: ***Model K2-I-c*** *representing the continuous relations for the eight variables* $\{E, e, F_0, f_0, SLPCC, slpcc, RMFCC, rmfcc\}$.

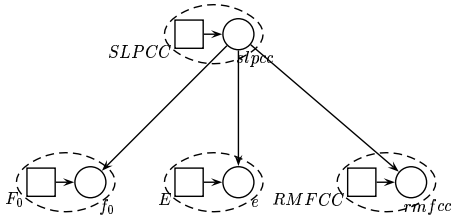


Figure 7.9: ***Model MDL-I-c*** *representing the continuous relations for the eight variables* $\{E, e, F_0, f_0, SLPCC, slpcc, RMFCC, rmfcc\}$.

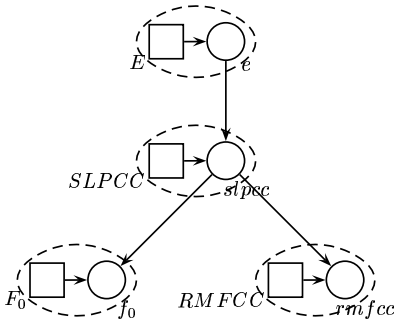


Figure 7.10: ***Model MDL-II-c*** *representing the continuous relations for the eight variables* $\{E, e, F_0, f_0, SLPCC, slpcc, RMFCC, rmfcc\}$.

In the first experiences, the number of components in the MGs to model the variables is as follows: 8 for the $SLPCC$, 8 for the $RMFCC$, 3 for F_0 (one for unvoiced parts) and finally 2 for$E$. The *LBG* [Linde *et al.*, 1980] algorithm was used to determine the initial setting for the Gaussian parameters. CPDs were learned with EM [Murphy, 2001; Bilmes and Zweig, 2002].

Results with the three systems (Models **K2-I-c**, **MDL-I-c** and **MDL-II-c**) are shown in Figure 7.11.Table 7.1 shows the EER score for the three systems.

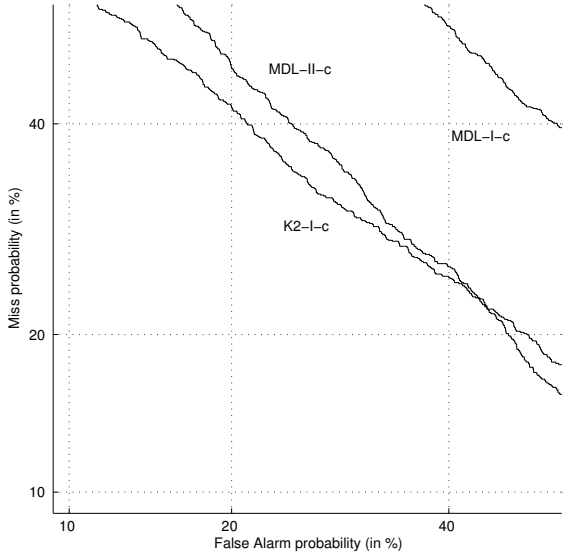


Figure 7.11: *Results obtained with the three structures using the continuous relations.*

Those results show the influence of the structure (conditional independencies) in the final score. It can be seen that results do not agree with the MDL results obtained in the structure research. After discretization step the best structure using the MDL approach was **MDL-I**. Then, this results would not be normal. To explain this result the key word is discretization. As it was already explained the obtained conditional independencies using the MDL approach are between region of the acoustical space.

Table 7.1: EER scores obtained with the three structures.

	K2-I-c	**MDL-I-c**	**MDL-II-c**
score	30.51	45.21	31.91

To verify the reason explained in the previous paragraph some experiments were done. In the first one the variable E was eliminated of the **MDL-I-c** model. Figure 7.13 shows the results, EER = 29.17.

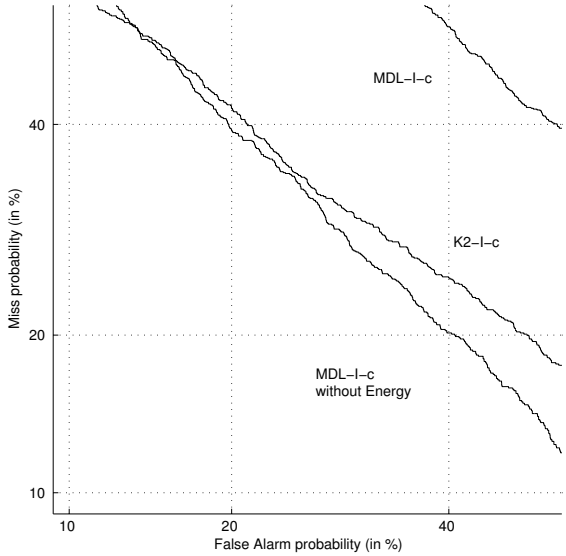


Figure 7.12: *Results obtained to verify the influence of E in the MDL-I-c model.*

The results obtained using this last structure shows that variable E has a bad influence into the structure of model **MDL-I-c**. To measure the influence of the relation of E in the second experience the relation between the E and $SLPCC$ variables was eliminated of the **MDL-I** model. Figure 8.5, where the EER = 29.07.

This last result shows that E has a very limited contribution to the final score. A possible justification to this fact is the few number of components in the mixture of gaussians used to represent the variable.

An important remark about BNs with continuous relations can be made comparing the obtained scores with that obtained using only the $SLPCC$ variable using the same number of components in the mixture of gaussians. Figure 7.14 and Table 7.2 show the results where it is seen that an improvement in the ERR is obtained. Also a contribution to the false acceptance rate is remarked.

Table 7.2: EER scores obtained with BNs and $SLPCC$

	K2-I-c	**MDL-I-c**	**MDL-II-c**	$SLPCC$
score	30.51	45.21	31.91	31.42

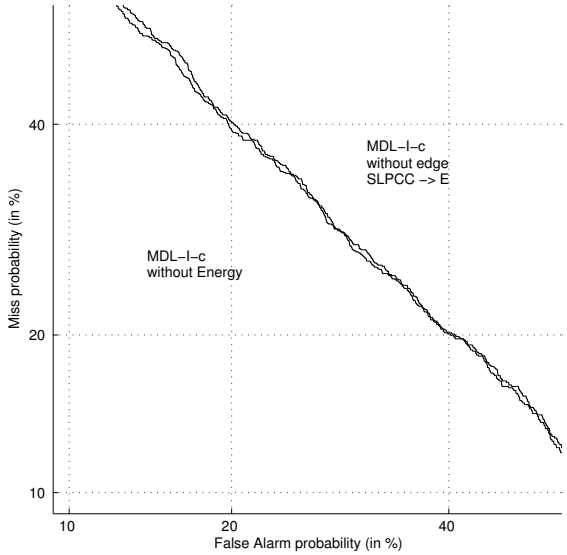


Figure 7.13: *Results obtained to verify the influence of edge $SLPCC$ -> E in the MDL-I-c model.*

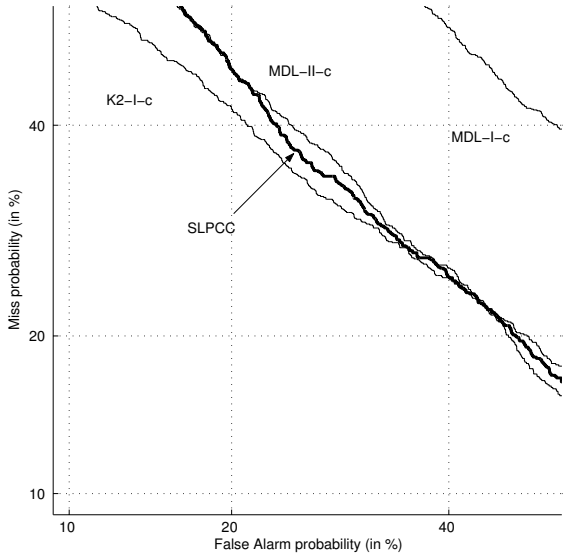


Figure 7.14: *Results obtained with BNs using the continuous relations and the Spectral information $SLPCC$.*

7.6.2 Using the Discrete Relations

The same set of experiments was done using the relations between the discrete variables as is represented in Figure 7.6, depicted again for clarity. This model will be called from now **Model K2-I-d**, d for discrete, Figure 7.15. Now the models in Figure 7.16 is called **Model MDL-I-d**.

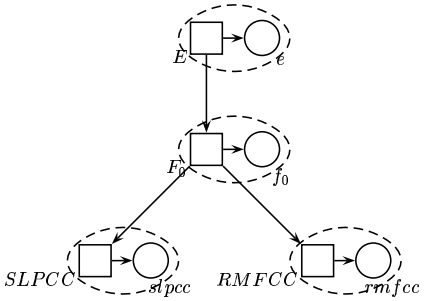


Figure 7.15: ***Model K2-I-d*** *with a relation between the discrete variables which determine the used gaussian in each GMM.*

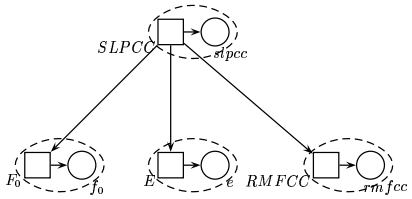


Figure 7.16: ***Model MDL-I-d*** *with a relation between the discrete variables which determine the used gaussian in each GMM.*

Results obtained with those structures are shown in Figure 7.17 and Table 7.3 .

Table 7.3: EER scores obtained with the three structures.

	K2-I-c	**MDL-I-c**	**MDL-II-c**
score	27.48	26.64	28.27

Again, the difference between the structures is reflected in the scores. Now, the best score is obtained with the structure of model **MDL-I-d**. The behavior shows by this structure agree with the MDL approach used to learn the conditional independence in the model.

7.6.3 Using Dynamic Bayesian Networks

We know now that prosodic information has to do with features whose domain are more longer units that just frames of 20 ms. Also, we know that prosodic features structure the flow of speech into the time. Then, we think that uses some dynamics into this kind of features can help to improve the SV system performance. In this experiment we uses just the spectral information $SLPCC$ and F_0 where the pitch has an edge which

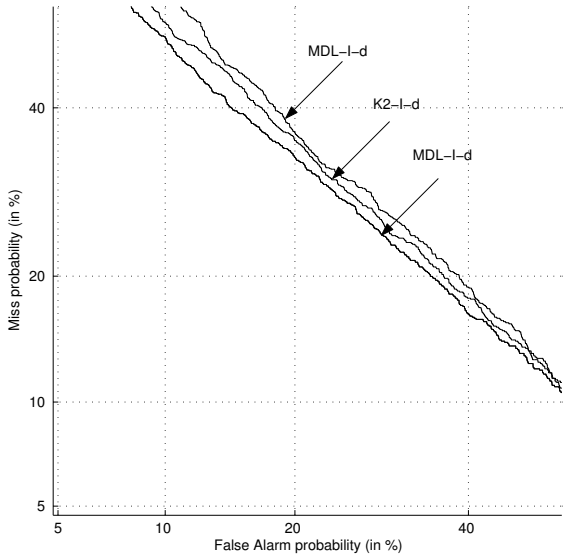


Figure 7.17: *Results obtained with the three structures (**K2-I-c**, **MDL-I-c**, **MDL-II-c**) using the discrete relations.*

relate the variables into the time, figure7.18. for this experience the tool kit GMTK [Bilmes and Zweig, 2002] was used. The upper index represent the time in each variable.

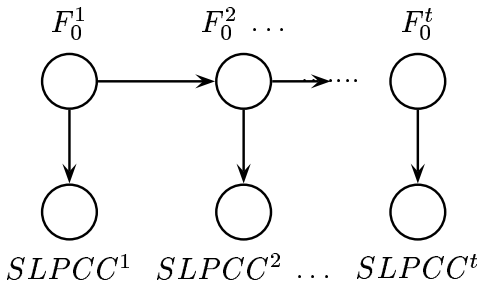


Figure 7.18: *Dynamc Bayesian Network modeling the prosodic information with a temporal edge.*

Figures 7.19 shows the results, EER = 26.41, with this dynamic structure. This result shows that dynamic can improve the performance the performance of using just $SLPCC$. Using just The same variables without the dynamic part the performances are almost the same.

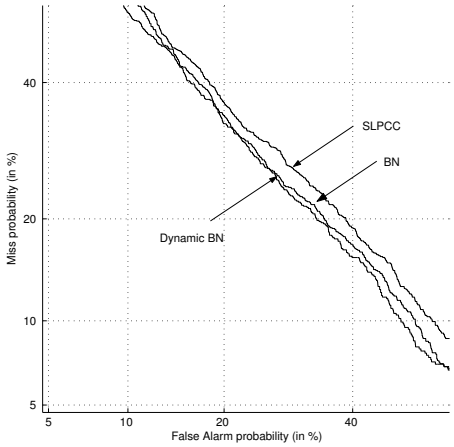


Figure 7.19: *Experiment III, DET curves for System based on Dynamic BNs.*

7.7 Conclusions

This chapter presents the techniques used in one of the basic contributions of this work. We have proposed to use BN for SV as a statistical model to combine different aspects of speech keeping its relations of conditional independencies. The structure used to combine those aspects were learned using the techniques of structure learning presented in this chapter. The two basic approaches to learn the structure in a BN were presented.

Parameters learning was the other aspect presented in this chapter. Principles and techniques for parameters learning were reviewed. The classic maximum likelihood approach as well as the Expectation Maximization technique were presented. The latter uses, for Bayesian Networks, in the expectation step the inference techniques developed in the previous chapter. This fact comes to justify the importance of the last chapter.

With this chapter we started to show how to obtain the models used in the proposed SV system, but there remains the adaptation of those models. This process is described in the next chapter where an a priori is used to compute the parameters.

Chapter 8

Models Adaptation

Adaptation is the process by which something, in this case a model, changes or is changed so that it can be used in a different way or in different circumstances. From this point of view model adaptation techniques are a way to reduce the problems of mismatch between the conditions in the training and test phases. A particular case is the adaptation of generic model to an specific model when a small amount of adaptation data is collected from this particular model. Here the initial conditions are observed in training the generic model and the particular conditions are those observed in the collected adaptation data. In general, the objective of model adaptation techniques is to adjust the parameters of a specific model using new data.

8.1 Classical Approaches

There are many kind of adaptation techniques but in general, from a classic classification, two main broad approaches exist, namely direct and indirect adaptation. This classification is reestablished by [Mokbel, 2001] where a unified view is proposed.

Indirect model adaptation is a transformation-based technique. The parameters of the model, all or by part, are transformed using a unique shared function, for example Maximum Likelihood linear regression (MLLR) adaptation. The hyper parameters for this function are computed using the set of adaptation data or new data. This technique is well adapted to the case when the available quantity of new data is small given that each sub set of parameters is transformed simultaneously and in general, all the parameters are transformed. Unlike indirect adaptation techniques direct ones do not use any functional transformation but try to reestimate the new model parameters as is done in Maximum a posteriori (MAP) technique. Therefore, just the parameters for which new data are available are locally adapted.

In the field of ASV several methods exist for models construction and in consequence several methods for its adaptation. Some methods use statistical models like Hidden Markov Models (HMM) [Matsui and Furui, 1994] and Gaussian Mixture Models (GMM) [Gauvain and Lee, 1994], where either the variances, or the means or both are adapted. Other techniques are Dynamic Time Warping (DTW)[Naik and Doddington, 1986] or Neural Networks [Mistretta and Farrell, 1998]. The first adaptation technique for ASV were proposed by Reynolds [Reynolds, 1997]. However, model adaptation remains a very important problem in ASV systems.

8.1.1 Maximum Likelihood Linear Regression

At the begin this adaptation technique [Leggetter and Woodland, 1995] was proposed for HMM where the emission distribution is modeled using GMM and just the means are adapted. Consequently it can also be applied to plain GMM since it can be modeled as a HMM with only one state. The new parameters for the adapted model are computed by a linear combination of the parameters in the original model. For a GMM with N components in a SV (Speaker Verification) context just the means (μ_i) are adapted and weights (w_i)

and variances (Σ_i) are taken directly from the original model. Then, the new means $\hat{\mu}_i$ are computed in the following way :

$$\hat{\mu}_i = \mathbf{A}_i \mu_i + \mathbf{b}_i,$$

where the matrix **A** and the vector **b**, the hyper parameters, are obtained by maximizing the likelihood of the new data. This maximization can be done using a modified **EM** algorithm. This solution needs a matrix inversion and that could be a problem because the small amount of data could gives an ill conditioned one. Therefore in practical cases it might be better to uses the generalized EM algorithm. The update equations for **A** and **b** are :

$$\mathbf{A}_i = \mathbf{A}_i + \lambda \frac{\partial Q}{\partial \mathbf{A}_i}, \tag{8.1}$$

$$\mathbf{b}_i = \mathbf{b}_i + \lambda \frac{\partial Q}{\partial \mathbf{b}_i}, \tag{8.2}$$

where λ is a learning rate and the partial derivatives are computed as follows :

$$\frac{\partial Q}{\partial \mathbf{A}_i} = \sum_{t=1}^{T} p(i|x_t) \frac{x_t - \mathbf{A}_i \mu_i - \mathbf{b}_i}{(\sigma_i)^2} \mu_i, \tag{8.3}$$

$$\frac{\partial Q}{\partial \mathbf{b}_i} = \sum_{t=1}^{T} p(i|x_t) \frac{x_t - \mathbf{A}_i \mu_i - \mathbf{b}_i}{(\sigma_i)^2}, \tag{8.4}$$

where $p(i|x_t)$ is the posterior probability of data x_t for a given gaussian i, and σ_i is the diagonal of Σ_i.

As a result of this algorithm the parameters of the new model are close to the parameters of the source model. A problem issue of hyper parameters computation is generated when just a small amount of new data is available. Now, with the same date the values for the matrix **A** of two times the observations dimension d and the vector **b** of dimension d should be computed. To overcome this problem some parameters could share the same hyper parameters.

8.1.2 Maximum a Posteriori

In the Bayesian approach [Gauvain and Lee, 1994] the parameters θ are assumed to be random variables with a priori distribution $p_0(\theta)$. Therefore, if the Bayes' rule is employed the posterior probability of θ given a set of observations **X** can be written in the following equation :

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})}.$$

Therefore, the calculated parameter $\hat{\theta}$ are those which maximize the posterior probability density of θ:

$$\begin{aligned} \hat{\theta} &= arg\max_{\theta} p(\theta|\mathbf{X}) \\ &= arg\max_{\theta} p(\mathbf{X}|\theta)\ p(\theta). \end{aligned} \tag{8.5}$$

The obstacle at a first view is to specify a prior for $p(\theta)$. In [Gauvain and Lee, 1994] it is proposed to a Dirichlet density for the weights of the mixture and a normal-Wishart density for the means and standard deviations because their conjugate distributions are well adapted to the problem. For example, the Dirichlet distribution is the conjugate of a multi-normal distribution which is ideal for the weights. Once again, from experience is well known that just mean adaptation is important to achieve good performance. In

[Gauvain and Lee, 1994] the update equation for the means is obtained and expressed in the next equation :

$$\hat{\mu}_i = \frac{\alpha_i \ \mu_i \ + \sum_{t=1}^{T} p(i|x_t) x_t}{\alpha_i + \sum_{t=1}^{T} p(i|x_t)},$$

where T is the number of observations x_t and i the index of Gaussians. The α value depends on the Gaussian and is chosen by cross-validation.

A modified version of this update equation is proposed in [Reynolds, 2000]. The difference between both equations is just the α factor :

$$\hat{\mu}_i = \alpha_i \mu_i + (1 - \alpha_i) \frac{\sum_{t=1}^{T} p(i|x_t) x_t}{\sum_{t=1}^{T} p(i|x_t)}.$$

Here α is computed as follows :

$$\alpha_i = 1 - \frac{\sum_{t=1}^{T} p(i|x_t)}{r + \sum_{t=1}^{T} p(i|x_t)},$$

where r is called relevance factor which is also computed by cross validation.

8.2 Bayesian Networks Adaptation

To be able to solve the parameters estimation some assumptions are made. The first is that the graph structure (G) is known and fixed. It should be remembered that the graph in a BN is a directed Acyclic Graph (DAG). The second assumption is that data are Identical Independently Distributed (i.i.d.), of which T cases are observed:

$$x = \{x(1), \ldots, x(t), \ldots, x(T)\}. \tag{8.6}$$

And the most important assumption, at this moment, is that all variables are observed. This condition is also known as complete observability.

By Bayesian theorem approach the unknown parameters, called θ, are treated as random variables. This way of proceeding permits to write the next equation:

$$P(\theta|x) = \frac{P(x|\theta) \ P(\theta)}{P(x)}, \tag{8.7}$$

Given that x are observed, the $P(x)$ can be seen as a normalization constant. Then, the optimal parameter values $\widehat{\theta}$ are obtained as follows:

$$\widehat{\theta} = arg\max_{\theta} P(x|\theta) \ P(\theta). \tag{8.8}$$

In this equation the prior distribution characterizes the prior knowledge and statistics of parameters. In general the choice of this distribution is a very important problem, which can be based on the experience,

the data itself or on mathematical reasons. This last factor in some cases is the most important. In our case the prior distribution is selected in such a way that the posterior distribution and the prior belong to the same family. In our case, the prior distribution corresponds to the distribution in the world model. If this condition is verified the prior distributions is called the conjugate distribution.

If x are i.i.d., as it is already said, the likelihood for all the observed cases is computed in the next form:

$$P(x|\theta) = \prod_{t=1}^{T} P(x(t)|\theta), \tag{8.9}$$

and then:

$$\widehat{\theta} = arg\max_{\theta} \prod_{i=1}^{N} P(x_i|\theta)\ P(\theta). \tag{8.10}$$

If, as is supposed, we work with a complete observability database each Conditional Probability Density (CPD), the parameters for the BN, can be trained independently from each other. Then, the log-likelihood for a BN with N different variables $\{x_1, \ldots, x_n, \ldots, x_N\}$ and T observed cases is written in the next form:

$$\begin{aligned} \log P(x|\theta) &= \log \prod_{t=1}^{T} \prod_{i}^{N} P(x_i(t)|Pa(x_i(t)), \theta_i) \\ &= \sum_{t=1}^{T} \sum_{i}^{N} \log P(x_i(t)|Pa(x_i(t)), \theta_i). \end{aligned} \tag{8.11}$$

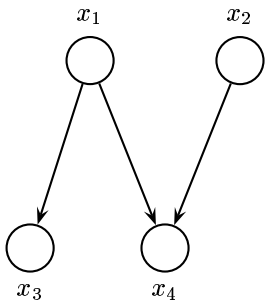


Figure 8.1: *BN to represent the probability distribution on equation 8.11.*

For example, if the structure is that one show in Figure 8.1 the probability of variables $x = \{x_1, x_2, x_3, x_4\}$ given the parameters θ can be written in the following form:

$$P(x|\theta) = P(x_1|\theta_1)\ P(x_2|\theta_2)\ P(x_3|x_2, \theta_3)\ P(x_4|x_1, x_2, \theta_4). \tag{8.12}$$

Then, given the previous hypothesis, the parameters θ can be learned from the four separated networks, one for each variable with its parents, as is show in Figure 8.2.

8.3 Parameters Estimation for Discrete BN

The first approach is to use Maximum Likelihood Estimation (MLE). For discrete variables each Conditional Probability Table (CPT) is modeled as a multinomial distribution. For this distribution the parameters are written in the following form:

$$\theta_{ijk} = P(x_i = j|Pa(x_i) = k), \tag{8.13}$$

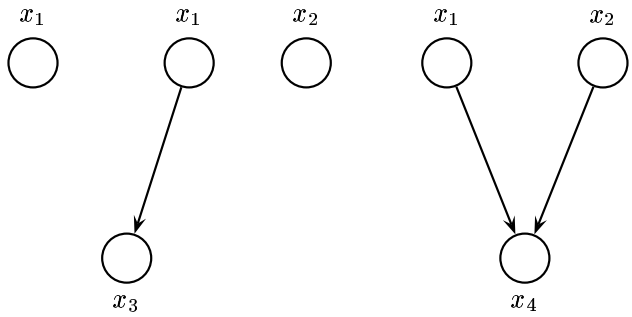


Figure 8.2: *Four Sub-graphs structures obtained from graph in Figure 8.1 for learning given that the variables are all observed. The four CPD, one for each subgraph, can be learned just taking into account the variables in the subgraph.*

where θ_{ijk}, the actual parameter, represents the probability that the variable x_i is in the state j and its parents $Pa(x_i)$ in the state configuration k.

For a multinomial distribution the sufficient statistics are just counting of possible configurations in each CPT. Then, if M samples are observed, this counting is computed as is represented in the next equation:

$$N_{ijk} = \sum_{t=1}^{T} \mathbf{1}_{(x_i(t)=j, Pa(x_i(t))=k)}. \tag{8.14}$$

Therefore, the last equation let us write the log likelihood as follows:

$$\begin{aligned} \mathcal{L} &= \log \prod_{m}^{M} \prod_{ijk}^{N} \theta_{ijk}^{N_{ijk}} \\ &= \sum_{m}^{M} \sum_{ijk}^{N} N_{ijk} \log \theta_{ijk}, \end{aligned} \tag{8.15}$$

where, as it is known, the sum of all possible state j probabilities for a variable x_i verify the constraint $\sum_j \theta_{ijk} = 1$. Then, the MLE optimization problem can be solved using the Lagrange multiplier and the mentioned constraint.

But, MLE does not take advantage of available prior knowledge. Then, if some prior information is available a Bayesian approach could be more suitable as is expressed in Equation 8.10.

To obtain the posterior probability distribution with the wished form we should look for appropriate a prior distribution, that is, the conjugate of a multinomial distribution. The solution in this case is the Dirichlet distribution, which will be used as the a prior in the Bayesian approach. The Dirichlet distribution has the form shown in the next equation:

$$P(\theta|\alpha) = \frac{1}{Z(\alpha)} \prod_{i=1}^{N} \theta_{ijk}^{\alpha_{ijk}-1}, \tag{8.16}$$

where $Z(\alpha)$ is a constant. This constant is function of the Gamma function with parameters α:

$$\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} \exp^{-t} dt. \tag{8.17}$$

Then, for N observed variables $x = \{x_1, \ldots, x_n, \ldots, x_N\}$, the likelihood can be evaluated as follows:

$$P(x|\theta_{ijk}) = \prod_{i=1}^{N} \theta_{ijk}^{N_{ijk}}, \tag{8.18}$$

and the prior distribution as a function of α values:

$$P(\theta_{ijk}|\alpha) = \frac{1}{Z(\alpha)} \prod_{i=1}^{N} \theta_{ijk}^{\alpha_{ijk}-1}, \tag{8.19}$$

and finally, the posterior distribution is written as follows:

$$P(\theta_{ijk}|x, \alpha) = \frac{1}{Z(\alpha)} \prod_{i=1}^{N} \theta_{ijk}^{N_{ijk}+\alpha_{ijk}-1}. \tag{8.20}$$

Now, the problem is remained to an optimization problem where the optimal $\widehat{\theta}_{ijk}$ parameters are the unknown. This problem can be established as follows:

$$\begin{aligned} \widehat{\theta}_{ijk}^{MAP} &= arg\max_{\theta_{ijk}} P(\theta_{ijk}|x, \alpha_{ijk}) \\ &= arg\max_{\theta_{ijk}} \prod_{i=1}^{N} \theta_{ijk}^{N_{ijk}+\alpha_{ijk}-1}, \end{aligned} \tag{8.21}$$

where $\frac{1}{Z(\alpha)}$ has been eliminated because is a constant. Using the log likelihood instead of just the likelihood the equation becomes:

$$\widehat{\theta}_{ijk}^{MAP} = arg\max_{\theta_{ijk}} \log \prod_{i=1}^{N} \theta_{ijk}^{N_{ijk}+\alpha_{ijk}-1}, \tag{8.22}$$

with the constraint given by the next equation:

$$\sum_{j=1}^{N} \theta_{ijk} = 1. \tag{8.23}$$

Therefore, we can use the Lagrange multiplier to maximize the function. The Lagrange multiplier is defined for this function as follows:

$$L(\theta_{ijk}, \lambda) = \sum_{i=1}^{N} (N_{ijk} + \alpha_{ijk} - 1) \log(\theta_{ijk}) + \lambda(\sum_{j=1}^{N} \theta_{ijk} - 1). \tag{8.24}$$

Taking the derivative for θ_{ijk}:

$$\frac{\partial\, L}{\partial \theta_{ijk}} = \frac{N_{ijk} + \alpha_{ijk} - 1}{\theta_{ijk}} = 0, \tag{8.25}$$

and for λ:

$$\frac{\partial\, L}{\partial \lambda} = \sum_{j=1}^{N} \theta_{ijk} - 1 = 0, \tag{8.26}$$

we obtain the searched optimal value:

$$\begin{aligned}\widehat{\theta}_{ijk}^{MAP} &= \frac{N_{ijk}+\alpha_{ijk}-1}{\sum_{j=1}^{N} N_{ijk}+\alpha_{ijk}-1} \\ &= \frac{\sum_{j=1}^{N} N_{ijk}}{\sum_{j=1}^{N} N_{ijk}+\alpha_{ijk}-1}\frac{N_{ijk}}{\sum_{j=1}^{N} N_{ijk}} + \frac{\sum_{j=1}^{N} \alpha_{ijk}-1}{\sum_{j=1}^{N} N_{ijk}+\alpha_{ijk}-1}\frac{\alpha_{ijk}-1}{\sum_{j=1}^{N} \alpha_{ijk}-1}, \quad (8.27)\end{aligned}$$

where:

$$\frac{\sum_{j=1}^{N} N_{ijk}}{\sum_{j=1}^{N} N_{ijk}+\alpha_{ijk}-1} + \frac{\sum_{j=1}^{N} \alpha_{ijk}-1}{\sum_{j=1}^{N} N_{ijk}+\alpha_{ijk}-1} = 1, \quad (8.28)$$

if $\rho_{ik} = \frac{\sum_{j=1}^{N} N_{ijk}}{\sum_{j=1}^{N} N_{ijk}+\alpha_{ijk}-1}$, then $1-\rho_{ik} = \frac{\sum_{j=1}^{N} \alpha_{ijk}-1}{\sum_{j=1}^{N} N_{ijk}+\alpha_{ijk}-1}$ and finally we can write the next equation:

$$\widehat{\theta}_{ijk}^{MAP} = \rho_{ik}\frac{N_{ijk}}{\sum_{j=1}^{N} N_{ijk}} + (1-\rho_{ik})\frac{\alpha_{ijk}-1}{\sum_{j=1}^{N} \alpha_{ijk}-1}. \quad (8.29)$$

Then, the optimal value for the parameter of the model is a linear combination of the prior parameter α_{ijk} and of the counting in the new data N_{ijk}. From the coefficients, ρ_{ik} and $1-\rho_{ik}$, used to weight the prior parameters and the counting in the new data we can deduce an important remark. The more observations, $\sum_{j=1}^{N} N_{ijk}$, there are, the more important the first coefficient is ρ_{ik} and then, the second coefficient $1-\rho_{ik}$, which weights the prior information is negligible. In a similar way, the less observation there are, the more importance to the prior α_{ijk} is given.

The coefficients in the linear combination ρ_{ik} are function of i and k. That is, there is a coefficient for each value of the actual variable x_i and for the values taken by its parents k. If we remember that a CPT is a stochastic matrix, where each line corresponds to each value of the variable x_i and each column to the value of its parent, it is easy to see that such combinations is made between columns of stochastic matrix. This fact will be used for look for a measure which can gives an idea of the difference between two different models and then, to compute the ρ_{ik} factor.

8.4 Using Adapted Discrete Relations in Speaker Verification

Each CPT for discrete variables,describes the interaction between a node and its immediate predecessors [Sánchez-Soto *et al.*, 2004b]. Specifically CPTs represent the relationships between parents and a child. The child's value depends on the combination of values taken by its parents. If the relation to be adapted links two discrete variables a CPT adaptation can be done as it is just explained. Then, in this section we explore this possibility with some experiences.

The combination of both models (UBM and model learned with available data of each speaker) is based on a coefficient ρ_{ik} computed the available information or as it was already introduced (8.3, equation 8.29) using a distance measure between vectors of both CPTs. A CPT is the transition matrix which has for elements real numbers in the closed interval [0, 1] and has entries in a discrete finite Field F such that the sum of elements in each column/line equals to 1. Therefore, each column/line is a probability distribution for the corresponding instantiation of the parents in the local conditional dependence represented in the CPT. Each PDF in the CPT can be modified in order to change the relation between the variables or to modify the relation of all the variables in the BN.

For example, if all the variables are binary $X = \{x, \neg x\}$, the values $t_{i,j} = p(x = i|Pa(x) = j)$ in the CPT for the term $P(F_0|E)$ in (7.27) could be:

Table 8.1: CPT for $P(F_0|E)$.

	f_0	$\neg f_0$
e	0.45	0.35
$\neg e$	0.55	0.65

In a SV system a model called World model or Universal Background Model (M_m) is learned using a great quantity of data by hoping that the general characteristics of speakers can be well collected in the parameters of this model. This quantity of information is then adapted using the data from each speaker (s_i). In this way one obtains each final speaker model (M_{s_i}). Then, models which depend directly on the initial world model and the new data D_{s_i} for each speaker are obtained as follows:

$$M_{s_i} = F(M_m, D_{s_i}). \tag{8.30}$$

In BN the proposed technique to adapt the $CPTs$ could be based on the fact that each CPT is a stochastic matrix since verifies:

$$\begin{aligned} t_{i,j} \geq 0 \, \forall i,j \in B, \\ \sum_{i \in B} t_{i,j} = 1, \end{aligned} \tag{8.31}$$

where B is the set of conditioning values by column. Each column of this matrix is a stochastic vector which under certain conditions is a good approximation of the probability density function (pdf). Model adaptation involves estimating the new vectors in the matrix with a transformation that includes vectors in the world model and the speaker model.

Any modification to values in the pdf function implies necessarily modification to the dependencies between the variables modeled by RB. Then, this function can be used to perform adaptation. In this case the problem is brought back to a problem of comparison between two pdfs. On the basis of equation (8.30), CPTs of the final model will be a function of the CPT of the world model and the CPT obtained for the speaker before the adaptation:

$$CPT'_{s_i} = F(CPT_m, CPT_{s_i}), \tag{8.32}$$

where CPT'_{s_i} is the final model, CPT_m is the world model and CPT_{s_i} the model before adaptation. The adaptation using a combination of both initial CPTs is possible. A linear combination, that verifies the conditions in (8.31), as we have already seen in equation 8.29, is:

$$CPT'_{s_i} = \rho CPT_m + (1-\rho) CPT_{s_i}, \tag{8.33}$$

where $\rho \in [0\,1]$. The values for ρ could be fixed values as an approximation or can be also approximated by obtaining a suitable distance between both $pdfs$.

In a first test (**Experiment I**) the structure **Model K2-I** was used. A discretization of all variables was achieved in order to simplify the probabilistic scheme. Variables $SLPCC$ and $RLPCC$ were discretized using 32 values, E with two and F_0 with three values (one value for unvoiced part). In a second experiment the structures **K2-I-d** and **MDL-I-d** were also tested.

8.4.1 Experiment I

In this experiment the structure **Model K2-I** was used. A discretization of all the variables was achieved first. These experiments were done in a part, only 600 test segments) of the NIST Speaker Verification

database 2004. Each test segment is evaluated against 11 hypothesized speakers. A first test was made using a fixed value ρ for all speakers and a discrete model. Then two different distances were employed with the same discrete model. From (8.31), the $t_{i,j}$ values are subject to a unit sum constraint like proportions in a compositional data. Then, in Aitchinson geometric [Aitchison, 1986] structure of probability functions on finite intervals (a, b), a distance can be defined for any two pdf, f and g like:

$$d_A(f,g) = \left[\frac{1}{4L}\sum_{x=a}^{b}\sum_{y=a}^{b}\left(\log\frac{f(x)}{f(y)} - \log\frac{g(x)}{g(y)}\right)^2\right]^{1/2}. \tag{8.34}$$

where $L \in [a, b]$ is the interval's length, this distance verifies $0 \leq d_A(f, g) \leq 1$.

The second used distance was the Kullback-Leibler symetric distance that is just $d_K(f, g) \geq 0$, but in the experiments it has never been bigger than 1 given that both pdf are close to each other :

$$d_K(f,g) = \sum_{x=a}^{b} f(x)\log\frac{f(x)}{g(y)}, \tag{8.35}$$

where $[a, b]$ is the interval's length.

The performance of the system with these conditions is shown in the DET plot in Figure 8.3, where it can be seen the influence of ρ. The value for the fixed ρ is equal to 0.9. The Kullback-leibler distance shows the best performance.

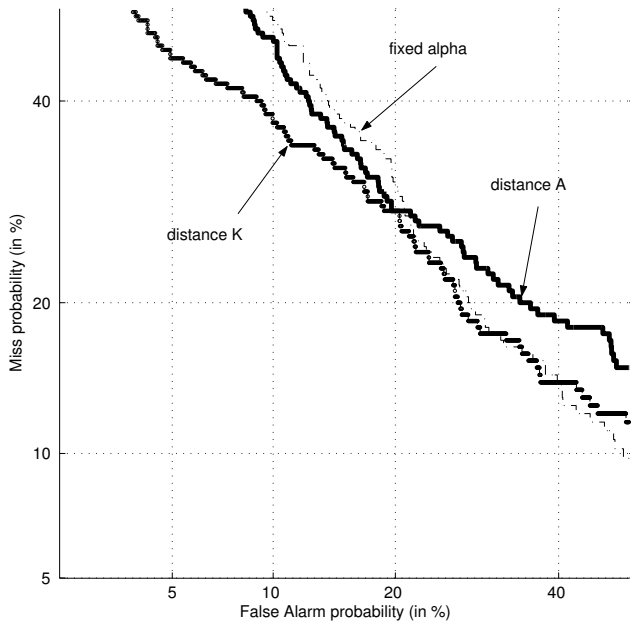


Figure 8.3: *DET Curves of the discrete system using a fixed ρ. ρ was obtained from Aitchinson distance d_A and Kullback-Leibler symmetric distance d_K.*

8.4.2 Experiment II

In this part will be shown the results obtained adapting the discrete relations in the structures **K2-I-d** and **MDL-I-d**. Results are present in the Figure 8.4 and table 8.2 depict the results.

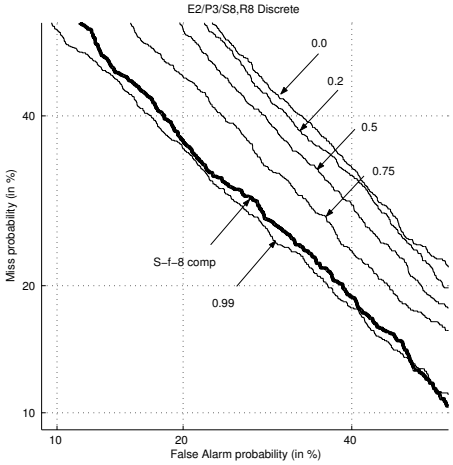


Figure 8.4: *Experiment II, DET curves for Discrete Relations Adapted in Structure $K2-I-d$.*

Table 8.2: *Experiment II, DET curves for Discrete Relations Adapted in Structure $K2-I-d$.*

ρ	0.0	0.2	0.5	0.75	0.99
score	36.68	35.81	34.08	31.71	27.48

Figures 8.5 and table 8.3 depict the results using the Kullback-Leiber and Variation Distance to adapt the CPTs.

Table 8.3: *Experiment II, DET curves for Discrete Relations Adapted in Structure $K2-I-d$.*

ρ	0.99	Kull	Vari
score	27.48	31.61	33.09

Figures 8.6 and table 8.4 depict the obtained results with the structure **MDL-I-d**. As well as in the previous experiments the discrete relations were adapted.

These results show that adapting the discrete relations do not contribute to improve the performance of the SV system. The conditional Probability tables learned with the UBM are better representation of the variable relations.

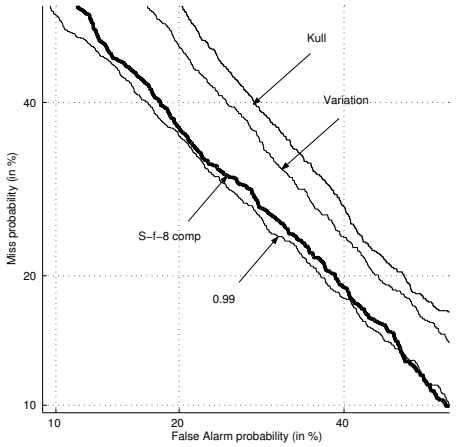


Figure 8.5: *Experiment II, DET curves for Discrete Relations Adapted in Structure $K2-I-d$.*

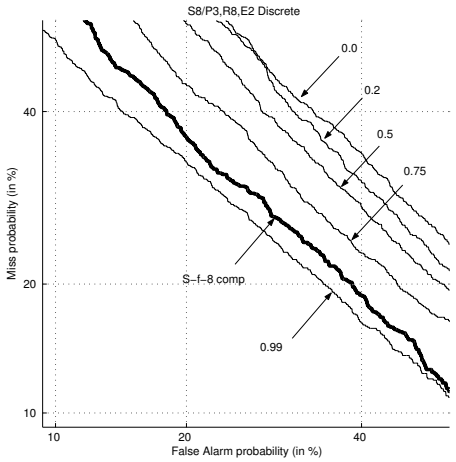


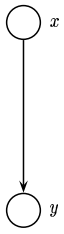
Figure 8.6: *Experiment II, DET curves for Discrete Relations Adapted in Structure $MDL-I-d$.*

Table 8.4: *Experiment II, DET curves for Discrete Relations Adapted in Structure $MDL-I-d$.*

ρ	0.0	0.2	0.5	0.75	0.99
score	37.33	35.64	33.78	30.92	26.64

8.5 Parameters Estimation for Continuous BN

Now the parameter estimation for two continuous variables will be developed. First of all, the distribution to model a continuous variable will be a normal distribution. Then, the relationship between two such variables is represented on the graph (Figure 8.7).



$$p(y|x) \backsim \mathcal{N}(y; m_y + \mathbf{B}(x - m_x), \mathbf{r}), \quad (8.36)$$

Figure 8.7: *Continuous BN (or Gaussian Network). Graphical representation of two continuous variables related by a conditional independence.*

for $y \in R^d$ and $x \in R^k$, where the regression matrix $\mathbf{B} \in R^{k \times d}$ and $\mathbf{r}$ is the precision matrix, the inverse of the covariance matrix Σ.

To use the Bayesian approach, in the same way that for the multinormal distribution, it should be found the conjugate of a Gaussian distribution. We know that for a Gaussian the conjugate distribution is also a Gaussian if only the mean is a variable. Otherwise, if the variance is also a variable the adequate distribution is the normal-Wishart.

Then, for a Gaussian distribution where the relationship between two continuous variables is presented the next equation can be written:

$$p(y|x, m_y, m_x, \mathbf{r}, \mathbf{B}) \propto |\mathbf{r}|^{1/2} \exp -\frac{1}{2}(y - m_y - \mathbf{B}(x - m_x))^t \mathbf{r}(y - m_y - \mathbf{B}(x - m_x)), \quad (8.37)$$

where m_i is the mean of i and $\mathbf{B}$ is the regression matrix that models the dependence between y and x.

The normal-Wishart distribution is defined as follows. Let x_i for $i \in [1, m]$ has a multivariate normal distribution of dimension p with null mean and covariance matrix Σ, and X is the matrix of dimension $m \times p$. Then, the matrix $X^t X$ of dimension $p \times p$ follows a Wishart distribution with scale matrix Σ and degree of freedom m. The joint prior for a precision matrix $\mathbf{r}$ is a Wishart distribution $W(\alpha, \mathbf{u})$ given by:

$$W(\alpha, \mathbf{u}) = C(\alpha, \mathbf{r})|\mathbf{r}|^{\frac{\alpha - p}{2}} \exp \frac{-1}{2} tr(\mathbf{ur}), \quad (8.38)$$

where, $C(\alpha, \mathbf{r})$ is a normalization constant, $\mathbf{u}$ a symmetric matrix, $\alpha > 1$ a effective sample size [DeGroot, 1970] and p is the dimensional of the space. Then the normal-Wishart distribution is the joint of a Wishart and a normal distribution $N(\mu, \tau\mathbf{r})$ as follows:

$$p(x, m_x, m_y, \mathbf{r}, \mathbf{B}|\tau, \mu, \alpha, \mathbf{u}) \propto |\mathbf{r}|^{\frac{\alpha - p}{2}} \exp -\frac{\tau}{2}(m_y + \mathbf{B}(x - m_x) - \mu)^t \mathbf{r}(m_y + \mathbf{B}(x - m_x) - \mu) \exp -\frac{1}{2} tr(\mathbf{ur}). \quad (8.39)$$

Using both equations, equation 8.37 as the likelihood and the normal-Wishart 8.39 as the prior distribu-

tion, the posterior distribution can be written as follows:

$$p(m_x, m_y, \mathbf{r}, \mathbf{B}|y, x, \tau, \mu, \alpha, \mathbf{u}) \propto |\mathbf{r}|^{\frac{\alpha-p+1}{2}} \quad \exp -\frac{1}{2}(y - m_y - \mathbf{B}(x - m_x))^t \mathbf{r}(y - m_y - \mathbf{B}(x - m_x)) \exp -\frac{\tau}{2}(m_y + \mathbf{B}(x - m_x) - \mu)^t \mathbf{r}(m_y + \mathbf{B}(x - m_x) - \mu) \exp -\frac{1}{2} tr(\mathbf{ur}) \tag{8.40}$$

The maximum a posteriori (MAP) of this function can be computed using the log of it. Then, we obtain the next equation:

$$\log p(m_x, m_y, \mathbf{r}, \mathbf{B}|y, x, \tau, \mu, \alpha, \mathbf{u}) \propto \frac{\alpha - p + 1}{2}|\mathbf{r}| \quad -\frac{1}{2}(y - m_y - \mathbf{B}(x - m_x))^t \mathbf{r}(y - m_y - \mathbf{B}(x - m_x)) -\frac{\tau}{2}(m_y + \mathbf{B}(x - m_x) - \mu)^t \mathbf{r}(m_y + \mathbf{B}(x - m_x) - \mu) -\frac{1}{2} tr(\mathbf{ur}) \tag{8.41}$$

To maximize this equation the gradient technique can be employed. Therefore we compute the partial derivative for $\mathbf{B}$ using the next fact:

$$\frac{\partial}{\partial \mathbf{M}}(\mathbf{M}a + b)^t \mathbf{C}(\mathbf{M}a + b) = (\mathbf{C} + \mathbf{C}^t)(\mathbf{M}a + b)a^t.$$

After this computation, we have the next equation:

$$\frac{\partial}{\partial \mathbf{B}} \log p(m_x, m_y, \mathbf{r}, \mathbf{B}|y, x, \tau, \mu, \alpha, \mathbf{u}) = -\mathbf{r}(\mathbf{B}(m_x - x) + y - m_y)(m_x - x)^t -\tau \mathbf{r}(\mathbf{B}(x - m_x) + m_y - \mu)(x - m_x)^t = 0.$$

If the equation is pre multiplied by $(-\mathbf{r}^{-1})$ we obtain the next equation where the total posterior probability for all the observed cases k is represented as follows:

$$\sum_k (\mathbf{B}(m_x - x^k) + y^k - m_y)(m_x - x^k)^t + \tau \sum_k (\mathbf{B}(x^k - m_x) + m_y - \mu)(x^k - m_x)^t = 0.$$

We can still arranging the equation:

$$\sum_k \mathbf{B}(m_x - x^k)(m_x - x^k)^t + \sum_k (y^k - m_y)(m_x - x^k)^t + \tau \sum_k \mathbf{B}(x^k - m_x)(x^k - m_x)^t + \tau \sum_k (m_y - \mu)(x^k - m_x)^t = 0.$$

and:

$$\sum_k \mathbf{B}(m_x - x^k)(m_x - x^k)^t + \tau \sum_k \mathbf{B}(x^k - m_x)(x^k - m_x)^t + \sum_k (y^k - m_y)(m_x - x^k)^t + \tau \sum_k (m_y - \mu)(x^k - m_x)^t = 0.$$

since $(\mathbf{A} - \mathbf{B})^t = (\mathbf{A}^t - \mathbf{B}^t) = -1\ (-\mathbf{A}^t + \mathbf{B}^t) = -1\ (\mathbf{B} - \mathbf{A})^t$ we have:

$$\sum_k \mathbf{B}(m_x - x^k)(m_x - x^k)^t + \tau \sum_k \mathbf{B}(x^k - m_x)(x^k - m_x)^t - \sum_k (y^k - m_y)(x^k - m_x)^t - \tau \sum_k (\mu - m_y)(x^k - m_x)^t = 0.$$

if $(\mathbf{A}-\mathbf{B})(\mathbf{A}-\mathbf{B})^t = (\mathbf{B}-\mathbf{A})(\mathbf{B}-\mathbf{A})^t$:

$$\mathbf{B}\sum_k((x^k-m_x)(x^k-m_x)^t+\tau(x^k-m_x)(x^k-m_x)^t) - \sum_k((y^k-m_y)(x^k-m_x)^t+\tau(\mu-m_y)(x^k-m_x)^t)=0,$$

then:

$$(1+\tau)\mathbf{B} \quad \sum_k((x^k-m_x)(x^k-m_x)^t = \sum_k((y^k-m_y)(x^k-m_x)^t+\tau(\mu-m_y)(x^k-m_x)^t).$$

Finally we obtain the optimal value for the regression matrix $\widehat{\mathbf{B}}$ which is:

$$\widehat{\mathbf{B}} = \frac{1}{(1+\tau)}\left(\sum_k(y^k-m_y)(x^k-m_x)^t\right)\left(\sum_k(x^k-m_x)(x^k-m_x)^t\right)^{-1} + \frac{\tau}{(1+\tau)}\left(\sum_k(\mu-m_y)(x^k-m_x)^t\right)\left(\sum_k(x^k-m_x)(x^k-m_x)^t\right)^{-1}, \tag{8.42}$$

where we can see that the coefficients, or weights sum one:

$$\frac{1}{1+\tau}+\frac{\tau}{1+\tau}=1, \tag{8.43}$$

if $\beta=\frac{1}{1+\tau}$, then $1-\beta=\frac{\tau}{1+\tau}$ and finally we can write the next equation:

$$\widehat{\mathbf{B}} = \beta\left(\sum_k(y^k-m_y)(x^k-m_x)^t\right)\left(\sum_k(x^k-m_x)(x^k-m_x)^t\right)^{-1} + (1-\beta)\left(\sum_k(\mu-m_y)(x^k-m_x)^t\right)\left(\sum_k(x^k-m_x)(x^k-m_x)^t\right)^{-1}, \tag{8.44}$$

The last equation shows how the a priori information μ present on the term $(\mu-m_y)$ is combined with the empirical information obtained from the samples y^k and present on term (y^k-m_y). Comparing this equation with that 7.16 obtained learning parameters for a known structure and full observability suing just MLE. This equation can be seen as a generalization where the contribution of the a priori model and the observed data is again function of their covariances.

This last developement consider the mean values of x and y, respectively m_x and m_y, as constant values. In each EM step the m_x and m_y values are changed. Thus, it should be established a procedure to adapt the three concerned values,m_x, m_y and B, in this dependence, see equation 8.36. Using the same procedure, the already established posterior distribution, equation 8.40, and the gradient technique, the optimal value for the mean of x, m_x, is obtained from the next equation :

$$\frac{\partial}{\partial m_x}\log p(m_x,m_y,\mathbf{r},\mathbf{B}|y,x,\tau,\mu,\alpha,\mathbf{u})=0.$$

If the derivative of second order, where $\mathbf{C}$ is assumed symetric with respect to a is :

$$\frac{\partial}{\partial a}(b-\mathbf{M}a)^t\mathbf{C}(b-\mathbf{M}a)=-2\mathbf{M}^t\mathbf{C}(b-\mathbf{M}a),$$

then, rearranged the log posteriori :

$$\begin{aligned}\log p(m_x, m_y, \mathbf{r}, \mathbf{B}|y, x, \tau, \mu, \alpha, \mathbf{u}) \propto \\ & \frac{\alpha - p + 1}{2}|\mathbf{r}| \\ & -\frac{1}{2}(y - m_y - \mathbf{B}x - \mathbf{B}m_x)^t\mathbf{r}(y - m_y - \mathbf{B}x - \mathbf{B}m_x) \\ & -\frac{\tau}{2}(m_y + \mathbf{B}x - \mathbf{B}m_x - \mu)^t\mathbf{r}(m_y + \mathbf{B}x - \mathbf{B}m_x - \mu) \\ & -\frac{1}{2}tr(\mathbf{ur}) \end{aligned} \tag{8.45}$$

its derivative with respect to m_x is expressed in the nex equation :

$$\begin{aligned}\frac{\partial}{\partial m_x}\log p(m_x, m_y, \mathbf{r}, \mathbf{B}|y, x, \tau, \mu, \alpha, \mathbf{u}) &= -\mathbf{B}^t\mathbf{r}\mathbf{B}m_x - \mathbf{B}^t\mathbf{r}(y - m_y - \mathbf{B}x) \\ & \quad -\tau\mathbf{B}^t\mathbf{r}\mathbf{B}m_x - \tau\mathbf{B}^t\mathbf{r}(\mu - m_y - \mathbf{B}x) = 0.\end{aligned}$$

then, we can write this last equation in the following form:

$$\begin{aligned}(-1-\tau)(\mathbf{B}^t\mathbf{r}\mathbf{B})m_x = \quad & \mathbf{B}^t\mathbf{r}(y - m_y - \mathbf{B}x) \\ & \tau\mathbf{B}^t\mathbf{r}(\mu - m_y - \mathbf{B}x)\end{aligned}$$

After some mathematical manipulations the optimal value of the mean m_x for all the observed cases k is as follows:

$$\begin{aligned}m_x = \quad & \frac{1}{-1-\tau}\mathbf{B}^{-1}\sum_k(y^k - m_y - \mathbf{B}x^k) + \\ & \frac{\tau}{-1-\tau}\mathbf{B}^{-1}\sum_k(\mu - m_y - \mathbf{B}x^k),\end{aligned}$$

where:

$$\frac{1}{-1-\tau} + \frac{\tau}{-1-\tau} = -1$$

Once the m_x value is adepted, the values for m_y and $\mathbf{B}$ can be computed. The steps to obtaine the optimal value of m_y is through the same manner of proceeding used in the previous part. Then, the rearranged log posteriori is:

$$\begin{aligned}\log p(m_x, m_y, \mathbf{r}, \mathbf{B}|y, x, \tau, \mu, \alpha, \mathbf{u}) \propto \\ & \frac{\alpha - p + 1}{2}|\mathbf{r}| \\ & -\frac{1}{2}(-m_y - \mathbf{B}(x - m_x - \mathbf{B}^{-1}y))^t\mathbf{r}(-m_y - \mathbf{B}(x - m_x - \mathbf{B}^{-1}y)) \\ & -\frac{\tau}{2}(m_y + \mathbf{B}(x - m_x - \mathbf{B}^{-1}\mu))^t\mathbf{r}(m_y + \mathbf{B}(x - m_x - \mathbf{B}^{-1}\mu)) \\ & -\frac{1}{2}tr(\mathbf{ur}) \end{aligned} \tag{8.46}$$

If the derivative of second order, where $\mathbf{C}$ is assumed symetric with respect to b is :

$$\frac{\partial}{\partial b}(b - \mathbf{M}a)^t\mathbf{C}(b - \mathbf{M}a) = -2\mathbf{C}(b - \mathbf{M}a),$$

we obtaine the following equation:

$$\begin{aligned}& -\mathbf{r}(y - \mathbf{B}(x - m_x)) + \mathbf{r}m_y + \\ & \tau\mathbf{r}(-\mu + \mathbf{B}(x - m_x)) + \tau\mathbf{r}m_y = 0\end{aligned}$$

and finally the optimal value for m_y is expressed in the next equation:

$$m_y = \frac{\tau}{1+\tau}\sum_k(\mu - \mathbf{B}(x^k - m_x)) + \frac{1}{1+\tau}\sum_k(y^k - \mathbf{B}(x^k - m_x))$$

where the addition of the coefficients is equal to:

$$\frac{1}{1+\tau} + \frac{\tau}{1+\tau} = 1$$

Finally, the process to adapt the BN is as follows. The serie of operations performed in each pair of dependent variables has to start by adapting the mean value of the parent, in the previous equations defined by m_x. This value will be used to compute the mean value of the child, m_y. Once both values have been obtained the parameter which define the relation between both variables, B, will be computed in the next step. For variables in between the roots and the final children, as soon as the values for a given pair of dependent variables have been obtained the mean for the child will be used to start again the process in the pair where this last variable is the parent. In a given structure, this procedure will starts by the root, in a tree, or by the roots, in a polytree and will ends with the final children.

8.6 Using Adapted Continuous Relations in Speaker Verification

As it was done for the adapted CPTs, this section present some results for adapted continuous relations. For the second set of tests (**Experiment III**) the three structures (**Model K2-I-c**, **Model MDL-I-c**, **Model MDL-II-c**) were used. The original variables were modeled by GMMs as was done in the previous section (7.5). $SLPCC$ and $RMFCC$ with 8 components, F_0 with five components and E with four components in order to establish the influence of components number in each GMM. In all these experiments, we should remember, the mean was adapted according to section 7.6 using a factor $\alpha = 0.75$.

8.6.1 Experiment III

Again the three structures already mentioned (**Model K2-I-c**, **Model MDL-I-c**, **Model MDL-II-c**) were used. Here the relationships between continuous variables were adapted.

Adaptation of CPD is made using fixed values (β). In the Figures it can be seen the evolution of EER score given that value. Special attention has to be paid to the values 0.0 and 1.0. The regression matrix used in learning is that of the world model if $\beta = 1.0$ and is learned just with the speaker data if $\beta = 0.0$.

Results obtained with the structure **MDL-I-c** are presented in Figure 8.8 and Table 8.5. The curves in the graph cross each other, then the legends are useless. Notice that this structure is really dependent of the adaptation factor β.

Table 8.5: EER scores obtained with the structure MDL-I-c

β	0.99	0.75	0.5	0.2	0.0
MDL-I-c	34.35	36.02	37.62	38.74	39.58

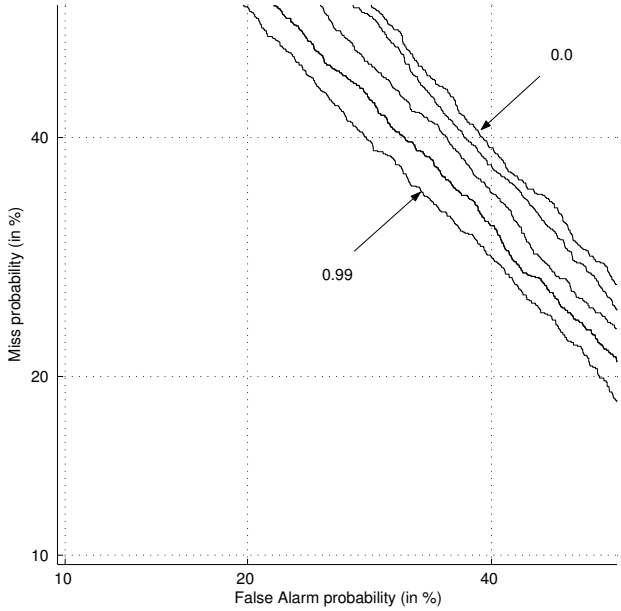


Figure 8.8: *Results obtained using the structure MDL-I-c in Figure 7.9.*

Results obtained with the structure **MDL-II-c** and the same protocol are presented in Figure 8.9 and Table 8.6.

Table 8.6: EER scores obtained with the structure MDL-II-c

β	0.75	0.5	0.2
MDL-II-c	30.85	30.8591	31.98

The most important in this graph is to notice the stability of this structure with respect to values of β and its influence in the FRR.

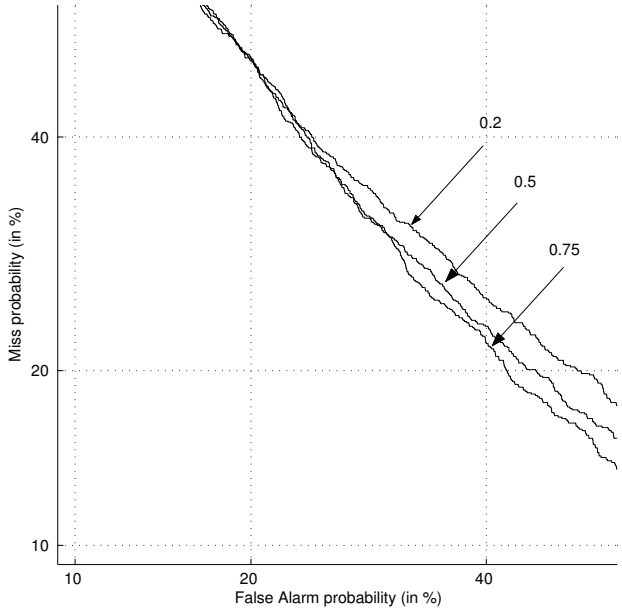


Figure 8.9: *Results obtained using the structure MDL-II-c in Figure 7.10.*

The same test was made using the structure **K2-I-c** obtained from the K2 algorithm. Results also with the same protocol are presented in Figure 8.10 and Table 8.7.

Table 8.7: EER scores obtained with the structure K2-I-c

β	0.99	0.75	0.2	0.0
K2-I-c	30.45	31.37	38.74	39.21

Again, the instability of this structure is reflected on the changes of values of β.

8.7 Conclusions

This chapter presents a basic contribution of this work. In order to obtain a robust SV system models adaptation should be done. This chapter presents the basic adaptation techniques used for models based on GMM at the beginning and the proposed techniques for BN and the end. We first explained the adaptation for multinomial distribution. Using a MAP technique we have computed the optimal parameters to combine the a prior Dirichlet distribution and the new data. The final equation shows that the parameters are a linear combination of the parameters represented in the Dirichlet distribution and the counting in the new data.

In the same way, we have obtained the optimal parameters for a multinormal distribution again with a MAP adaptation. This time the a prior used was a normal-Wishart distribution. The parameters are a combination of the prior distribution and the new data weighted by a parameter in the a prior distribution which depends on the used quantity of data.

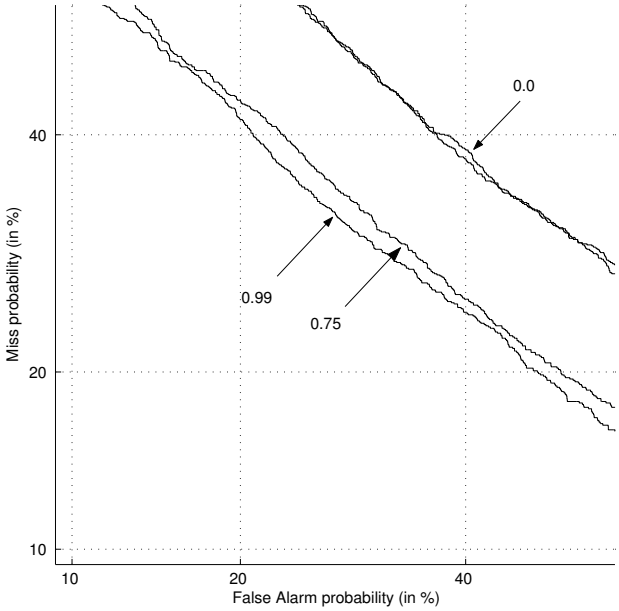


Figure 8.10: *Results obtained using the structure K2-I-c in Figure 7.8.*

Chapter 9

Conclusions and Perspectives

We have presented in this work an alternative to methods for Speaker Verification Systems based on Bayesian Networks. The objective of this works were to study and to propose a solution to combine different aspects of speech in a unique model. The effort was undertaken because there was a lack of resources which can help to combine in a robust way different aspects of speech keeping its characteristics and relations.

Not only basic Bayesian Networks were used but procedures to learn the relations between the variables and new procedures to adapt these relations were developed. Graphical representations of conditional independencies of different variables, such as energy and pitch, could not be easy modeled directly on a Graphical Model. We thus propose to learn the structure directly from the available data. Even if the latter was already an established technique in machine learning, but we diverted its usage and transformed it into a technique for representing and learning the physical relations between the concerned variables. The advantages of these graphical representation was to give us an easy physical interpretation of obtained results.

Through out this work we mainly emphasized a graphical and practical approach to look for the relations between the variables in a physical level. We believe that much processes in speech production are of a multivariate nature, where each variable have a relation or influence on each other. In the obtained structures it is often difficult to distinguish a precise independence relation between the variables, but this fact was due to the differents parameterized methods used. The most clear example is the structure obtained using the MDL approach. The found relations are between different acoustical regions because of the vectorial quantification made to provide the samples.

We showed how to build the structure, expliciting a mixture of gaussians to model each variable in a unique statistical model. The obtained results show that this approach can help to improve results obtained just using spectral information. Configurations appears to be the most important factor in the final performance. Each structure has shown a particular behavior even thought that all of them are using the same data and same initialization. This is partly due to the reflected relationships among the variables. A close look to those given relations was done to analize the results. A particular example was done to show the bad influence of the relations between the energy and the spectral information in one of the proposed models. Moreover, if care is not taken to measure the conditional independence relations the Bayesian Networks approach losses all its benfits.

After having tested two different configurations, relation between discrete variables and relation between continuous variables, it can be said that continuous configurations appears to give the most important performance. Even if the discrete relations have an influence into the used gaussians to model each varaible, the observed variables provide with must more information. Then, model the relations between the observations, or continuous variables, gives most important results.

Besides the purely analysis of the conditional relations between the variables, we tried to use prosodic information. We known prosodic information and suprasegmental characteristics, like intonation, accent or pitch are very important in a normal communication. Also, we know that its domain is not a single phonetic segment, but larger units of more than one segment, possibly whole sentences or even longer utterances. Then, we proposed Dynamic Bayesian Networks for representing the synchronical and dynamical properties of prosodic characteristics. Until now, the results are not conclusive because of the few experiments.

After having established the conditional independencies reflected in the structure of each Bayesian Network, we proposed and developed a technique to adapt those dependence relationships. The techniques is based on the Maximum a Posteriori (MAP) approach, where the a priori was obtained from a well trained model, called the world model. Two schemes were discussed, using discrete variables and using conbtinuous variables. For discrete relation, a procedure based on a measure between the Conditional Probability Tables (CPT), was proposed in addition to the found equation using MAP. This procedure takes advantage of the structure of each CPT. Given that a CPT is a stochastic matrix, each one of their vector is a probability density function (pdf). Then, a measure between two different pdf can be used to combine and, then, adapt a CPT. The obtained results, mainly using, a fixed weight to combine the CPTs shown that adaptation of discrete relations does not improve the performances. Adaptation of continuous relations are based on computing the values of a regression matrix. These values, as it is shown in the developed equation, depend on the covariance between the corresponding variables. The obtained results, using fixed values for the regression matrix, shows just a performance improuvement in a unique model.

This work demostrated that Bayesian Networks can be expected, if care is taken, to be a good approach to model differents aspects of speech in order to develop better Speaker Verification Systems.

9.1 Perspectives

Some question were left unanswered. The most important aspect to be continued, is the dynamic modeling of prosodic variables. We did not find an adequate approach to model the temporal relation between the variables. Consequently, a challenging task can be foresee.This modeling can include also, the periodicity of samples. We have not yet establish the searched relations between the acoustical variables and prosidic ones which can explain an approach to a suprasegmental modeling.

Another important question is regarding the investigation of source of information in a speech signal. It should be investigated the information in the residual signal. Maybe a better represented with another parameterization can be obtained.

In the conditional realtionships observed in the structure of a Bayesian Network, some fruitful perspectives are foreseen. A close study of conditional relations between each couple of variables could be done. We think about the example found in the energy and spectral information relation described before. Energy, modeled just with two gaussian components in a given structure, has not enough information to differentiate two different persons.

We did not study yet, the influence of the structure on each speaker. Alternatively, a research about the influence of the structure for each speaker should be done. Here we are talking about a structure adaptation. Of course, we can not pass over the increasing inportance of data. Adapting the structure can be a task more difficult because of the lack of data.

All the structures described have been tried but a combination of these structures could be explored in the future. A combine of clasifiers can improve the results. The potential of Bayesian Networks to combine variables in a single and unique statistical model. Given the obtained results, we think that a pursue of this work should be done in a practical approach, as well as, in theoretical questions.

Appendix A

NIST's Speakers Secognition Evaluation

Even if this part is presented in one appendix of this report, it was an important part, in time and effort, of the all work carried out during the PhD. "NIST's evaluation [NIST's Speaker Recognition Evaluation, 2004] is the internationally accepted standard for comparing speaker recognition technology and which forms the benchmark for measuring progress in speaker recognition research using conventional telephone speech" [1]. The objective of these yearly evaluations is to stimulate the researcher to exploring new ideas, which will be developed to be incorporated into new technologies, and, which will be finally tested in a well strict and established protocol.

The École Nationale Supéerieure des Télécommunications take part of the NIST' evaluation since 1998 as part of the ELISA consortium [The ELISA Consortium, 2004]. This consortium was created by three laboratories in France:

- **L**aboratoire d'**I**nformatique d'**A**vignon (LIA),
  http://www.lia.univ-avignon.fr/php/accueil.php
- **É**cole **N**ationale **S**uperiéure des **T**élécommunications (ENST),
  http://www.tsi.enst.fr
- **I**nstitut de **R**echerche en **I**nformatique et **S**ystèmes **A**léatoires (IRISA),
  http://www.irisa.fr/metiss/accueil.html

and then augmented by others laboratories, nowadays the next:

- Laboratoire de **C**ommunication **L**angagière et **I**nteraction **P**ersonne-**S**ystème (CLIPS),
  http://www-clips.imag.fr/geod
- University of Balamand - Lebanon,
  http://www.balamand.edu.lb
- Université de Fribourg, **D**ocuments, **I**mage and **V**oice **A**nalysis (DIVA) group,
  http://diuf.unifr.ch/diva/siteDIVA04/pages/home.xml

The Elisa consortium aims to facilitate the cooperation between the mentioned laboratories, mainly within the NIST's evaluation. The consortium organizes regular meetings, maintains a software platform [Alizé: a free, open tool for speaker recognition, 2004] for speaker recognition and help his members to be prepared for the international evaluation. The majority of the laboratories of the consortium take part jointly in the NIST's evaluations.

[1] Taken from Digital Signal Processing, January/April/July 2000

During my PhD work, as part of the ELISA consortium and the ENST, we participated to the NIST's evaluation in 2003 and 2004 in a close collaboration with Raphaël BLOUET and Gérard CHOLLET at the Signal and Images Department of the ENST and with Chafic MOKBEL at the University of Balamand [Blouet *et al.*, 2004]. My main contributions were in the secondary system year 2003 and primary and tertiary systems year 2004.

A.1 NIST's evaluation 2003

The year 2003 speaker recognition evaluation [NIST's 2003 Speaker Recognition Evaluation, 2003] had three main parts: one and two speakers detection and one speaker detection with extended data. In one speaker detection the training data for a target speaker is two minutes of speech from that speaker obtained from a single conversation. The test data is the only speech part of a single conversation obtained from one minute of conversation, speech is within 15 and 45 seconds. The difference of one speaker detection and two speakers detection tasks is that in the two speaker detection task the data, in training and test, contains the speech of two speakers. The training data has three whole conversations with both sides of the conversation summed together. The test data is a one minute segment from a single conversation with both sides of the conversation summed together again. Finally, the one speaker detection with extended data has almost one hour of speech for training. The test data is composed by all one side single conversation segments.

At the ENST we decided to perform just the one speaker detection task because of the resources needed to work with the two others tasks.

A.1.1 Primary system: ENST 1

The primary system is a segmental level linear combination of four GMM-based systems. Those systems are differing in the features vectors (FBCC and LPCC) and in the post-processing steps (feature warping and classical CMS). Speech segmentation is simply obtained with a GMM of Kc components, where $Kc = 64$.

In the test, block-level log-likelihood ratios followed by the T-norm in each system are calculated. Decision score is then obtain in two steps. We firstly apply a class by class fusion of the four systems scores. This permits to obtain Kc scores that we linearly combine to compute the final decision score. Systems fusion is obtained by applying logistic regression to each segmental scores. Class fusion is obtained by applying linear discriminant analysis.

Features

This primary system used the next parameters based sur the MFCC and LPCC acoustic representation:

- MFCC.1 40-dimensional features obtained as follows: 20-dimensional Mel-Frequency Cepstral Coefficients (MFCC), augmented by their first derivatives and warped to (0,1)-Gaussian distributions within 300-frames windows.
- MFCC.2 40-dimensional features obtained as follows: 20-dimensional Mel-Frequency Cepstral Coefficients (MFCC) with sliding cms, augmented by their first derivatives.
- LPCC.1 32-dimensional features obtained : 16-dimensional LP Cepstral Coefficients (LPCC, augmented by their first derivatives and warped to (0,1)-Gaussian distributions within 300-frames windows.
- LPCC.2 32-dimensional features obtained : 16-dimensional LP Cepstral Coefficients (LPCC), augmented by their first derivatives and warped to (0,1)-Gaussian distributions within 300-frames windows.

Modeling and Test

Two gender-dependent world models with 512 Gaussian components were created using the 2001 cellular development and evaluation datasets. The parameters of both world models were estimated using the EM algorithm. The target speaker models were obtained from the world model by adapting only the means. A total of 174 (100 female, 74 male) speaker models from the NIST-01 evaluation set served as cohort models to calculate the T-norm. Results obtained with this system by gender and location, to shows the diversity of the task, are shown in the next Figure A.1.

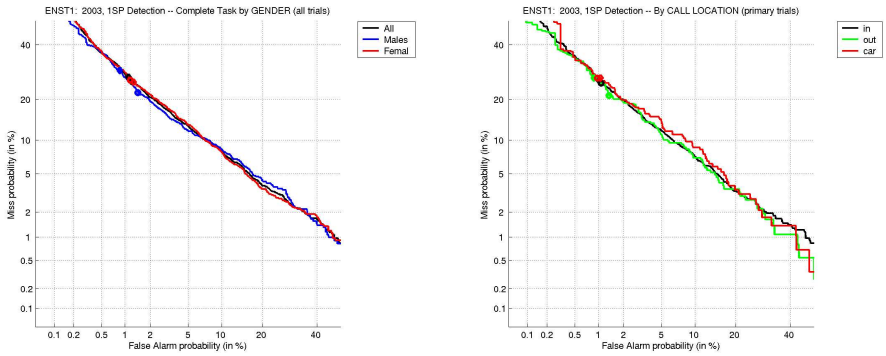


Figure A.1: *Results obtained for the primary system, NIST 2003*

A.1.2 Secondary system: ENST 2

The second system is based on Bayesian Networks (BN). We use BN in order to model the joint probability function of four set of features extracted form the speech signal. We use frame's energy (E), pitch (F_0), LPCC extracted from the speech ($SLPCC$) and from the LP-residual ($RLPCC$). Two GMM of 32 components permit to respectively model the two LPCC vectors sets. Two GMM of 2 components permit to model the pitch and the energy.

Features

A set of four types of parameters are extracted from the speech signal:

- $SLPCC$ 24-dimensional LP Cepstral Coefficients obtained as follow: 12-dimensional LPCC, with sliding CMS and augmented by their first derivatives.
- $RLPCC$ 24-dimensional LP Cepstral Coefficients obtained as in $SLPCC$ but extracted from the LP-residual.
- F_0 the frame pitch.
- E the frame energy.

Modeling and test

Two gender-dependent world models were created using part of the 2001 cellular development and evaluation datasets. Those data had been used with the K2 algorithm (section 7.1.1) to find the beast structure of the four variables. Issue of this analysis the conditional independence relations that define the network structure were obtained. This structure was set to be speaker independent, see next figure A.2:

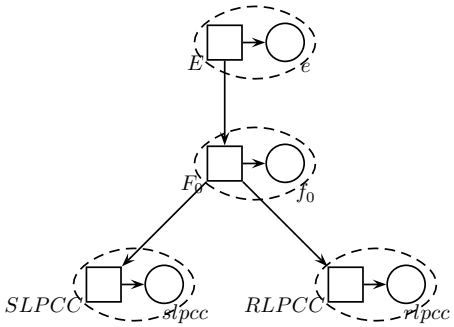


Figure A.2: *Structure of the Bayesian Network used for the Secondary system.*

The final world model parameters of the Bayesian Network (32/32/2/2 Gaussian components for $SLPCC$, $RLPCC$, F_0, E + conditional probability tables (CPT)) are then learned with the EM Algorithm using the GMTK toolkit [Bilmes and Zweig, 2002]. The final speakers models were obtained from the world model using two iterations of the EM algorithm. Just the means of the Gaussian were modified. CPTs from the world models were used for the speakers models. In the test, the decision score is directly based on the log-likelihood ratio without any kind of normalization. The results obtained with this system by gender and duration, again to shows the diversity of the NIST's evaluation, can be seen in the next figure A.3.

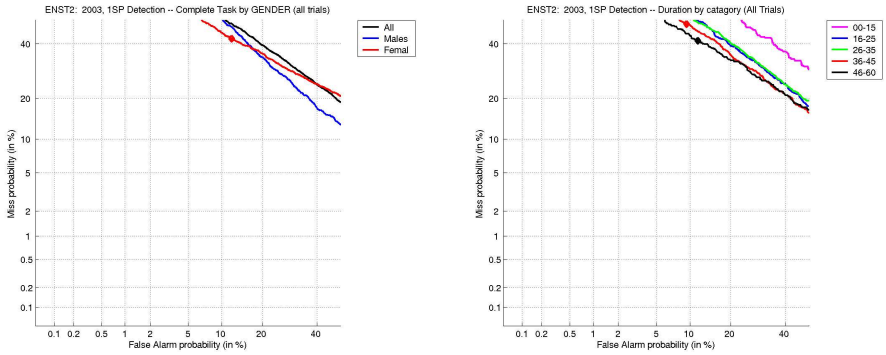


Figure A.3: *Results obtained for the primary system, NIST 2003*

A.1.3 Tertiary system: ENST 3

This system used a tree-based MLLR + smoothing using MAP algorithm. It had been mainly developed at the University of Balamand (Libanon), [Blouet *et al.*, 2004]. Training and test have been run at ENST.

Features

The same, already described, features used in the primary system MFCC.1 were used (see section A.1.1).

Modeling and Test

The speakers models were obtained by adapting the means of the gender correspondent world model using a tree-based MLLR + smoothing using MAP algorithm. The world models were created using the same

procedure as the primary system. The results are as follows figure A.4:

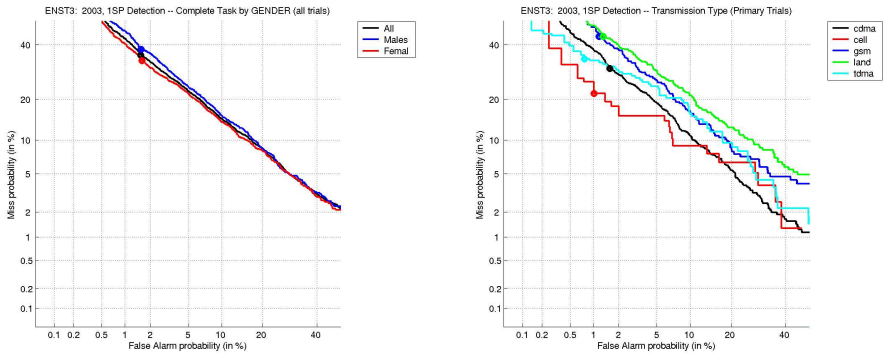


Figure A.4: *Results obtained for the primary system, NIST 2003*

A.2 NIST's evaluation 2004

Unlike the evaluation 2003, the evaluation 2004 [NIST's 2004 Speaker Recognition Evaluation, 2004] proposed only a one speaker detection task with 28 different conditions. These conditions are function of the available data for test and training. Training segments are continuous conversational speech without prior removal of intervals of silence. The seven training conditions are as follows:

- a single channel conversation side containing approximately 10 seconds of speech.
- a single channel conversation side containing approximately 30 seconds of speech.
- a single channel conversation side, of approximately 5 minutes total duration of speech and silence. [2]
- three single channels conversation sides involving the same speaker.
- eight single channels conversation sides involving the same speaker.
- sixteen single channels conversation sides involving the same speaker.
- sixteen single channels conversation sides involving the same speaker.
- three summed-channel conversation, formed by sample-by-sample summing of the two sides of the actual conversation.

The four test conditions are as follows:

- a single channel conversation side containing approximately 10 seconds of speech.
- a single channel conversation side containing approximately 30 seconds of speech.
- a single channel conversation side, of approximately 5 minutes total duration of speech and silence. [3]
- a single summed-channel conversation, formed by sample-by-sample summing of the two sides of the actual conversation.

[2] some files contains crosstalk
[3] some files contains crosstalk

An important characteristic of the database this years is the language spoken by the speakers. Most of the data involves English, but some conversations involves also bi-lingual speakers speaking Arabic, Mandarin, Russian and Spanish.

This year, the ENST has submitted results for the core test conditions (one-side-training-one-side-test) with three systems and also results for one-side-training-10-seconds-test and one-side-training-30-seconds with the primary system.

A.2.1 Primary system: ENST 1

This year the primary system was based on Bayesian Networks. The main difference between this system and that one of the year 2003 is the proposed adaptation of the dependencies between the conditioned variables (see section 8.2).

Modeling and test

The system uses a similar structure, figure A.2, where this time the relationship between the variables was established on the continuous variables. The set of parameters is the same used in the secondary system year 2003 (see section A.1.2).

The final world models parameters of the Bayesian Network use 16 Gaussian components for the $SLPCC$ and $RLPCC$ variables, 3 Gaussian components for the F_0 and 2 for the E. To model the probability relation between the continuous variables a regression matrix is used, see section 5.4. The final speakers models were obtained from the world model by adaptation of the Gaussian means and parameters of the regression matrix, see section 8.5 with a fixed value equal to 0.75 for all parameters. In the test, the decision score is directly based on the log-likelihood ratio without any kind of normalization. The results obtained with this system can be seen in the Figure A.5. The Figure A.6 shows the results obtained for the one-side-training-10-seconds-test and one-side-training-30-seconds conditions.

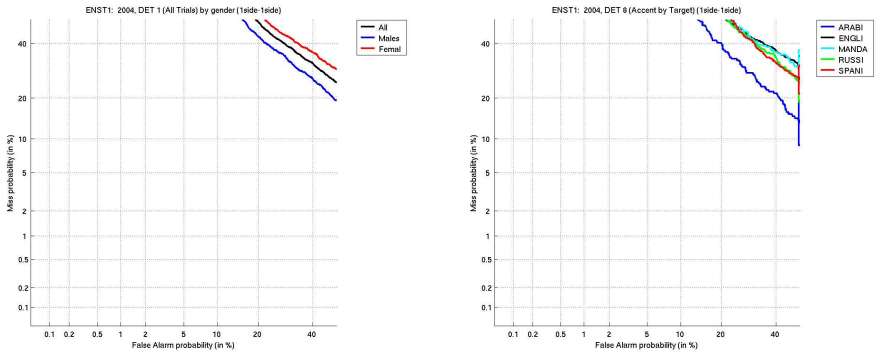


Figure A.5: *Results obtained for the primary system, BIST 2004. The BET curve in the left shows the results by gender (female, male and all). In the right side de BET curve shows the results by language.*

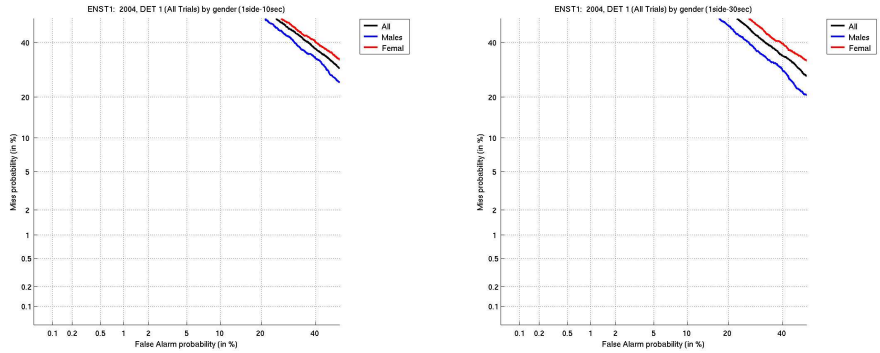


Figure A.6: *Results obtained for the primary system, BIST 2004, 10 seconds and 30 seconds test conditions. The BET curve in the left shows the results for the 10 seconds condition and the right BET curve shows the results for 30 seconds condition.*

A.2.2 Secondary system: ENST 2

The second system used a tree-based MLLE + smoothing using MAP algorithm, as the tertiary system in the year 2003 (see section A.1.3). It had been developed in collaboration with the University of Balamand (Lebanon). In the test, the decision score is directly based on the log-likelihood ratio without any kind of normalization. the results are shown in the next figure A.7:

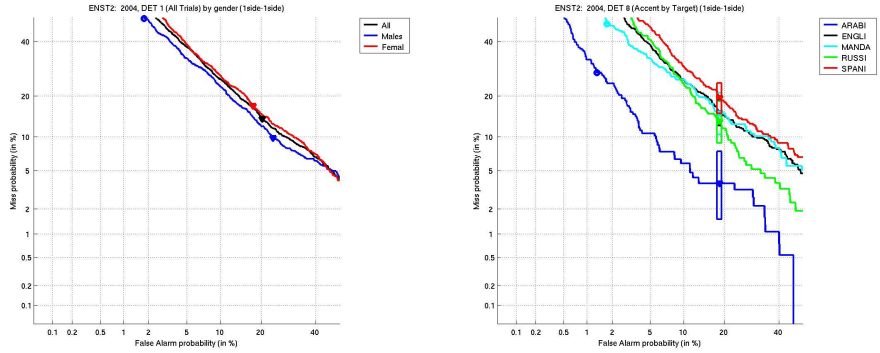


Figure A.7: *Results obtained for the secondary system, NIST 2004. DET curve in the left shows the results by gender (female, male and all). In the right side de DET curve shows the results by language.*

A.2.3 Tertiary system: ENST 3

This third system is based on the ALIZE platform [Alizé: a free, open tool for speaker recognition, 2004] and was part of the AGILE - ALIZÉ project. A sub-set of the primary system parameters were used and three GMM systems of 32 components were created. The first one uses the frame's voiced pitch values F_0. The second uses the LPCC extracted from the speech $SLPCC$ and the last system uses the LPCC from the LP-residual analysis $RLPCC$. A fusion was made based on a simple normalized scores addition.

Modeling and test

Two gender-dependent world models were created for each feature ($SLPCC$, $RLPCC$, and F_0) using part of the 2001 cellular development and evaluation datasets. The speakers models were obtained from the world models by adaptation. The means adaptation of the gender correspondent world model were made

by using MAP algorithm.

For all systems and given a test utterance, the decision score is the mean log-likelihood ratio between the target speaker and the world model over all the frames. The results are shown in figure A.8.

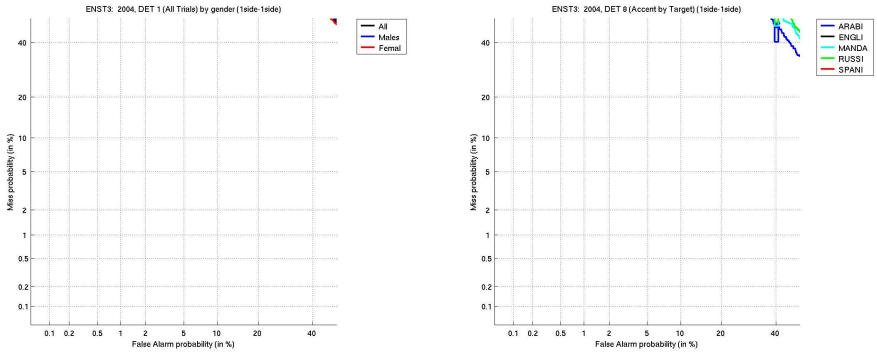


Figure A.8: *Results obtained for the tertiary system, NIST 2004. DET curve in the left shows the results by gender (female, male and all). In the right side de DET curve shows the results by language.*

ALIZE is not responsible for these results. This experience was carried out, first as a part of the AGILE - ALIZÉ project and second to verify in some way the potential of Bayesian Networks to combine different sources of information. Those results can be explained thinking on the systems' scores combination method, just a simple mean, and on the variables used in each system.

Appendix B

Publications

B.1 Workshop on Multimodal User Authentication. Santa Barbara, 2003

Eduardo Sánchez Soto, Raphaël Blouet, Gérard Chollet et Marc Sigelle.
Speaker Verification with Bayesian Networks.
Workshop on Multimodal User Authentication.
Santa Barbara, Californie. Decembre 11-12 2003.

Speaker Verification with Bayesian Networks

Eduardo Sánchez-Soto, Raphaël Blouet, Gérard Chollet and Marc Sigelle

École Nationale Supérieure des Télécommunications
Département de Traitement de Signal et des Images. LTCI/CNRS URA 820.
46 rue Barrault 75634 Paris Cedex 13 France
esanchez,raphael.blouet,gerard.chollet,marc.sigelle@tsi.enst.fr

Abstract

One solution to improve the performance of Speaker Recognition (SR) systems could be the integration of different aspects of the speech signal. Thus in this paper it is proposed to integrate, or fuse, all these informations in a probabilistic framework with a system based on Bayesian Networks (BNs) where the structure is learned directly from the data. BNs are a flexible and formal statistical framework that allows us to represent the conditional independence relations among different speech features that convey information about the speaker identity. In this paper, prosodic variables (pitch and energy), the linear prediction cepstral coefficients (LPCC) from signal and LPCC from residual signal of linear prediction analysis are used to represent each speaker.

This study is conducted on the NIST 2002 one speaker text-independent data base. These experiments confirm the potentialities of BN approach.

1. Introduction

Speech signal carries a lot of information besides the message. Other information about the speaker is present such as mood, emotive state and in particular his/her identity. SR (Identification (SI) or Verification (SV)) systems should use features which capture characteristics of the speaker in order to differentiate them from others. In this search for individual discriminant features some information could be lost. Many authors discard prosodic information in speaker verification, but it is known that they carry a lot of information about the speaker identity. Therefore speaker information of other sources must be used. The suprasegmental characteristics, like intonation, accent or pitch are really important in a normal communication, specially the pitch that appears like an important factor in speaker recognition [1]. However the pitch information in itself is not enough to discriminate between two different persons. Therefore speaker information of other sources must be used. For example, spectral information, conveyed by cepstral coefficients, and knowledge, which is not often taken in account, that comes from the source of excitation in speech production.

The main idea, developed in this paper, is to retrieve the conditional independencies directly from the data (linear residual analysis from the source in speech production, the spectral information from the vocal tract and prosody) in order to build a BN, and by this mean integrate, in a probabilistic way, all those informations.

This paper is organized as follows: Bayesian Networks are first introduced in section 2, with some discussion about the inference problem and algorithms. Section 3 reviews briefly some ideas about structure and parameters learning in BNs. In section 4, the experiments, results and their probabilistic interpretation are presented. Finally conclusions and perspectives are given in section 5.

2. Bayesian Networks

A BN, or Bayesian Belief Network [2], represents a joint probability distribution defined on a finite set of random variables. It is a formal representation, based on probability theory and graph theory, given by a Directed Acyclic Graph (DAG) in which nodes represent random variables and arcs represent conditional probabilistic dependencies among those variables. An arc from **Q** to **Y** can also be interpreted as indicating that **Q** has a direct influence on **Y**, Figure 1.

In a DAG each edge points from one node, called parent, to another, called child. In the same topology description, the node X_j has a descendant node X_i if this one is its child or is connected to it through its children. In a BN, a conditional probability distribution is associated with each node X_i that describes the dependency between this node and its parents, each node is conditionally independent from its non-descendants given its parents. Those dependence relations induces a factorization in the joint distribution function expressed as :

$$P(X_1, \ldots, X_N) = \prod_{i=1}^{N} P(X_i | Pa(X_i)), \qquad (1)$$

where $Pa(X_i)$ is the set of X_i's parents.

2.1. Inference

There are two main research problems in probabilistic reasoning using Bayesian Networks: learning and inference [3]. Bayesian network inference involves computing the posterior marginal probability distribution of some query nodes, and computing the most probable explanation given the values of some observed nodes once the structure is known.

A BN is a couple $(G, CPDs)$ formed by one structure, the graph G, and a set of Conditional Probability Distributions (CPD), one for each node with parents in the network. For nodes without parents we have just to specify their prior probability. Evidence, i.e. knowledge about the state of one variable, would modify the states of others variables in the network. Doing probability inference consists in computing

the probability of each state of a node when we know the state taken by some other variables. There are three types of evidence propagation, exact, approximated and symbolic. One or another is used depending on the characteristics of the data and the complexity of the structure. In order to make exact inference it is necessary to talk about "belief propagation" [4] and to take into account the relation of independence obtained directly from the graph. The exact methods present some problems. Some of them are not applicable to all the types of structures. The methods of general validity become very inefficient with certain structures when the number of nodes and its complexity grow. This is not surprising since it has been demonstrated that the exact propagation task is NP-hard [5]. For that reason, and from a practical point of view, the exact propagation methods can be very restrictive and even inefficient in situations in which the type of structure of the network requires a large memory and a lot of computational power. With the second method, approximated values are obtained using simulation methods as Monte Carlo and Gibbs sampling [6]. The last method of propagation works directly with symbolic parameters [7].

In general, if we have a set of variables $X = \{X_1, X_2, \ldots, X_N\}$ and a set E, the evidence, with known values $E = \{e_1, e_2, \ldots, e_M\}$, where $E \subset X$, inference consists in computing :

$$p(x_i|e) = \frac{p(x_i, e)}{p(e)} \; \alpha \; p(x_i, e). \tag{2}$$

The conditional dependence assumptions encoded by a BN have the advantage of simplifying the conditional probabilities computation. All this could be done in an equivalent tree structure when the original one is not a tree [8]. This structure is a tree built of cliques that represent the local structures, and then preserve the conditional probabilities. The first step in the junction tree construction consists in finding those cliques C_i. Then it is possible to compute their CPD. The CPDs of variables X_i are computed by marginalizing the cliques. In detail this process works has follows:

1. moralization and triangulation (because the parents are correlated given its children) of G to obtain an undirected graph G'.
2. computation of cliques C of G',
3. assign each X_i from X to one clique C_i,
4. for each $C_i \in C$ define a potential $\psi_i(C_i) = \prod_{X_i \in \mathcal{A}_i} P(x_i|Pa(x_i))$.

After those steps, the belief propagation method has to be applied to the new graph (collecting and distribution steps). That is, it must be updated the belief in each node when some variables have been observed.

3. Learning

The other main problem in probabilistic reasoning using Bayesian Networks is learning. Learning Bayesian Network from data [9] [10] consists in automatically constructing the network, structure and parameters, from information in data using some learning algorithms. The Statistical base of BN let the development of learning methods. We use these methods in order to obtain the conditional independences in the graph structure and the conditional probability distributions that quantify

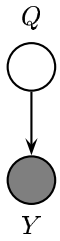


Figure 1: *Basic BN.*

those dependences directly from databases. Therefore, dependences, structure and conditional probability distributions can be learned from data.

3.1. Structure

In the process of finding the best structure, even if the space of variables is fully observable, some aspects must be considered. Firstly concerning the structure space, should trees be a priority or should more complex graphs be considered? The number of possible structures depends on the number of variables n in a super-exponential way. For example, with four variables there are 543 possible DAGs. It is unrealistic to explore all of them. For that reason, it has to be taken in consideration search algorithms that gives the structures to be evaluated. There are two different approaches to solve this problem, the first one, like MCMC [11], searches in all the structure space and returns either the best one, or the best in a Markov equivalent way. The second approach starts with a specific connected graph and then searches for independence relations in the data S, and puts in or takes away arcs.

The K2 algorithm [12], used in this work, belongs to the second approach. It starts with a structure, the simplest one, i.e. a graph without arcs. It needs some prior knowledge and a relationship between the variables. Then, for each variable X_i we look for the set $Pa(X_i)$. The variables in this set are restricted to those variables with smaller order numbers than X_i.

In order to achieve learning, a scoring function must be specified for measuring the network's quality. The criterion, or quality measure to select $Pa(X_i)$ is the last aspect to study in the structure learning. Maximum likelihood could be an adequate quality measure, but it privileges the fully connected graph. This graph gets the highest likelihood because it has the greatest number of parameters. Thus, to overcome this problem, a prior knowledge on the model can be used. By Bayes' rule, the MAP model is the one that maximizes :

$$P(G|S) = \frac{P(S|G)P(G)}{P(S)}, \tag{3}$$

where $P(G)$ penalizes complex model and $P(S)$ is a constant. The marginal likelihood is :

$$P(S|G) = \int_\theta P(S|G, \theta)P(\theta|G)d\theta, \tag{4}$$

where S is the database. (4) as the advantage that automatically penalizes more complex structures. This score function can be approximated [13] with a Laplace method, and finally get the BIC (Bayesian Information Criterion) :

$$logP(S|G) \approx logP(S|G,\hat{\theta}) - \frac{d}{2}logM, \qquad (5)$$

where M is the number of samples, $\hat{\theta}$ is the ML estimate of the parameters and d is the dimension of the model.

3.2. Parameters

Here, it is required to adjust the parameters of the BN in such a way that the CPDs describe the data statistically. The parameters θ and the model, $B(\theta)$, defined for these parameters are given. Also, the prior distribution over the models $P(B(\theta))$ and the space of parameters in these models $P(\theta|B)$ can be used. So, given some data S, it is wanted to estimate θ, such that the posterior probability to be maximized is :

$$P(B|S) = \frac{P(B)}{P(S)} \int_{\theta} P(S|\theta, B)P(\theta|B)d\theta. \qquad (6)$$

Thus the maximum likelihood estimate of θ is computed by minimizing the cost function over the probability density function. We can make an optimization that relies on the gradient of this function, or use an iterative procedure called Expectation - Maximization (EM) [14] or a variant, Generalized EM , using a gradient method in the M step.

4. Experiments and Results

In this section, experiments and results using our BN Speaker Verification System (BNSVS) are detailed.

4.1. Database

The data are taken from the second release of the Cellular Switchboard Corpus (Switchboard Cellular - Part 2) of the Linguistic Data Consortium (LDC) [15]. Each conversation is echo cancelled before use. The database is divided into training data (about 400 target speakers), and test data (about 3500 test segments). The training data for a target speaker consist in about two minutes of speech from that speaker, excerpted from a single conversation. Actual duration is, however, constrained to lie within the range of 110 to 130 seconds. Each test segment is extracted from a 1 minute excerpt of a single conversation and is the concatenation of all speech from the subject speaker during the excerpt. The duration of the test segment therefore vary, depending on how much the segment speaker spoke. So, the effective speech duration lies between 15 and 45 seconds. Both test and target speakers are of the same sex.

4.2. Modeling

The training and test parameter vectors consist of a set of four types of parameters. The first vector is a 24-dimensional LP Cepstral Coefficients obtained as follow : 12-dimensional LPCC, with sliding CMS (Cepstral Mean Substraction) and augmented with their first derivatives, $SLPCC$, for Signal Linear Prediction Cepstral Coefficients. The second vector, 24-dimensional LP Cepstral Coefficients has been obtained as before from the LP-residual signal $RLPCC$ [16][17], and finally the frame pitch F_0 and the frame energy E.

Those data had been used with K2 algorithm to find the best structure for our four variables. We have worked with all the possible orders and used the BIC score [5]. From this analysis we have obtained the conditional independence relations for the multivariate Gaussian distribution that define the network structure which is set to be speaker independent, Figure 2.

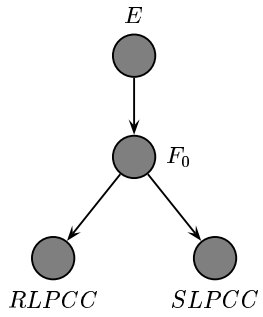


Figure 2: *Structure for the four variables (energy (E), pitch (F_0), signal $SLPCC$ and residual $RLPCC$) issued from the K2 algorithm.*

From basic probability theory the joint probability for the four variables $U = \{E, F_0, RLPCC, SLPCC\}$ can be written as:

$$P(U) = P(E)P(F_0|E)P(RLPCC|F_0, E) \\ P(SLPCC|F_0, E, RLPCC). \qquad (7)$$

Now, taking into account the graph of Figure 2 and its relations of conditional independence, this equation becomes a product of local terms :

$$P(U) = P(E)P(F_0|E)P(RLPCC|F_0) \\ P(SLPCC|F_0). \qquad (8)$$

The relation between $SLPCC$, $RLPCC$ and F_0 is obtained from the term $P(RLPCC|F_0)P(SLPCC|F_0)$. It can be interpreted as a relation of conditional independence where $RLPC$ and $SLPC$ are independent given F_0, noted $RLPCC \perp SLPCC|F_0$ or $I(RLPCC, SLPCC|F_0)$. Also, from the second term in (8) it can be seen that F_0 depends directly of E.

The physical interpretation of the relations between the variables gives the same relations found in the equations obtained from the graph. For example, the voiced speech has more energy that the unvoiced speech. It is evident that the speech energy depends directly from the speech voicing. This fact is written in the term $P(F_0|E)$. The source influences the spectral envelope due to the filtering effect of the vocal tract. The pitch is correlated with the vibration of the vocal folds and the vocal tract characteristics. Consequently, the source and the spectral envelope depends on pitch as it is seen in $P(RLPCC|F_0)P(SLPCC|F_0)$.

The relations obtained in equation (8) exhibit the causal interaction between the variables. Now, using Bayes theorem : $P(E)P(F_0|E) = P(E)P(F_0|E)$, the equation (8) can be rewritten as :

$$P(U) = P(F_0)P(E|F_0)P(RLPCC|F_0) \\ P(SLPCC|F_0). \qquad (9)$$

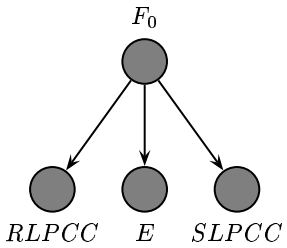


Figure 3: *Equivalent structure for the four variables (energy (E), pitch(F_0), signal $SLPCC$ and residual $RLPCC$) using the equality $P(E)P(F_0|E) = P(E)P(F_0|E)$.*

This new formulation corresponds to the graph shown on Figure 3. In this equation the causal relations represented are not similar to that presented in (8), but the probability density function is the same. Then the equation (9) also represents the variables relation. This structure has the advantage that pitch is the root node. Pitch is a feature whose domain is longer than just one single phonetic segment. Then the independence relations found in the equation (9) represent the conditional independence of $SLPCC$, $RLPCC$ and E given F_0. Where F_0 is a prosodic variable that relate different linguistic elements, by making boundaries and defining transitions in speech signal.

Once the structure has been learned, the final Universal Background Model (UBM) BN's parameters are learned. Since there are not enough training data for each speaker, adaptation methods are applied to compute every Target Speaker Model. For this purpose, the system starts from an universal model (UBM) which is then adapted to the client speaker by three iterations of the GEM algorithm and in this way we overcome the problem. Two gender-dependent UBM have been created using part of the 2001 cellular development and evaluation datasets (this database is similar to the database already described).

4.3. Results

Each test segment is evaluated against 11 hypothesized speakers. The decision score is directly based on the log-likelihood ratio between the target speaker and the UBM over all the frames without any kind of normalization. Figures 6 and 5 display the DET (Detection Error Tradeoff) curves that measure the performance obtained with our system and the standard technique Gaussian Mixture Models (GMM), that have become the dominant approach for modeling multivariate densities in text-independent speaker recognition. A DET curve is a mean of representing performances on detection tasks and is an standard in speaker and language recognition evaluations. In a DET curve, error rates are plotted on both axes (False Alarm and Miss Detection). It shows when a system fails to detect a target or declare such a detection when the target is not present.

First experiment uses the vector $SLPCC$ modelled by a GMM with 64 mixtures. The results shown in the DET curve, Figure 5, show a performance of 19.31 % at the Equal Error Rate (EER). The same has been done with the $RLPCC$ vector obtaining a score of 24.34 %. Now combining all the variables

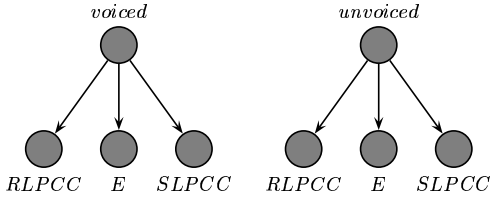


Figure 4: *Structure used in the second experiments.*

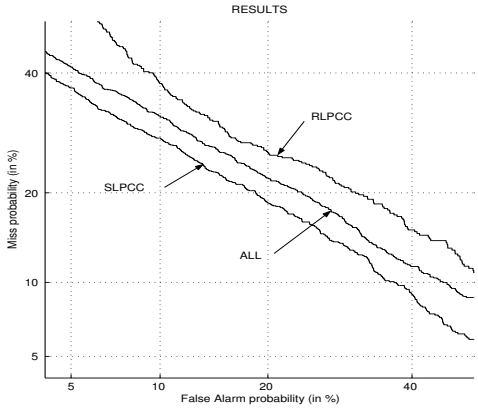


Figure 5: *DET curve for NIST 2002 evaluation data with SLPCC, RLPCC and All using a GMM with 64 mixtures.*

in a vector and using a GMM with 64 mixtures a 21.34 % score is obtained.

The next set of experiments use two models. The first one uses the structure in the Figure 2 and the set of parameters: 32 Gaussians for $RLPCC$ and $SLPCC$ plus 2 for the pitch F_0 and energy E. CPDs were learned with GEM [18] [19]. This choice of gaussian numbers (parameters number) was made taken into account the computation resources and time requests to finish a task. K-means was used to determine the initial setting for the Gaussian parameters. This system obtains an EER of 24%, Figure 6. The results in the Figures 5 and 6 show that a GMM with a $SLPCC$ vector perform better than our first system. Given that our score is similar to that obtained with the $RLPCC$ vector the difference can come from the independence relations obtained in the structure.

With the second structure shown in the Figure 4, a discretization of the continuous pitch F_0 was made in order to better modelize the voiced and unvoiced parts of speech. The parameters used for this model are : 2 values for the pitch (voiced and unvoiced), 16 Gaussians for the $RLPCC$ and $SLPCC$ and 2 Gaussians for the energy E. This system, shown in Figure 6 obtains an EER of 21.18% for male and 22.37% for female.

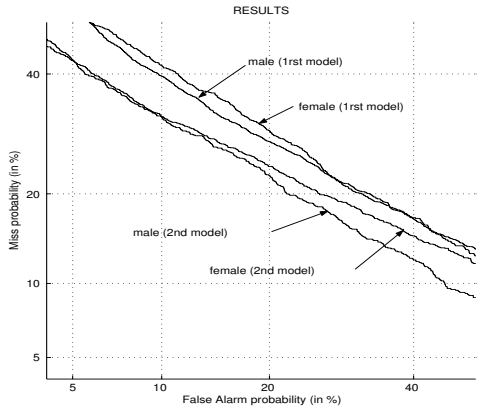


Figure 6: *DET curve for NIST 2002 evaluation data using our two Bayesian Network models: First Model as shown in Fig. 2 and Second Model as shown in Fig. 4.*

5. Conclusions and Perspectives

In this paper, a system achieving Speaker Verification based on BNs is presented. This system infers the Bayesian network structure automatically from the data. Also, it uses the independence relations obtained for integrating all the information presented on the speech signal in a single probability distribution. It shows that BNs are a flexible mathematical tool that can help to modelize information from different aspects of the speech signal. The physical interpretation given to the equations describing the structure suggests that the learning algorithms for BN are able to adequately infer the relations present in data. The perspectives for this work are important because of the flexibility of BNs. We expect further improvements from different research algorithms in the network structure learning and from the augmentation of parameters.

6. References

[1] Carey, M.J., Parris, E.S., Lloyd-Thomas, H., and Bennett, S., *Robust prosodic features for speaker identification*, International Confrence on Spoken Language Processing, vol.3, pp. 1800-1803, October 1996.

[2] Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.

[3] Murphy, K. P., *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. Dissertation, University of California, Berkeley, Fall 2002.

[4] Kim, K.P., and Pearl, J., *A Computational Model for Combined Causal and Diagnostic Reasoning in Inference System*, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) Morgan Kaufmann Publishers, San Mateo, CA, 190:193, 1983.

[5] Cooper, G.F., *The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks*, In Artificial Intelligence, 42:393-405, 1990.

[6] MacKay, D., *Introduction to Monte Carlo Methodes*, In M. Jordan, editor, Learning in Graphical Models, MIT Press, 1998.

[7] Castillo, E., Gutiérrez, J.M., and Haidi, A.S., *Parametric Structure of Probabilities in Bayesian Networks*, Lecture Notes in Artificial Intelligence, 956:89-98, 1995.

[8] Lauritzen, S.L., and Spiegelhalter, D.J., *Local Computations with Probabilities on Graphical Structures and their Applications to Expert Systems*, Journal of the Royal Statistical Society (1988), Series B, 50:157-224.

[9] Fisher, D., and Lenz, H.J., *Learning from Data: Artificial Intelligence and Statistics V (Lecture Notes in Statistics)*, Springer Verlag (Vol 112), New York, 1996.

[10] Castillo, E., Gutiérrez, J.M., and Hadi, A.S., *Expert Systems and Probabilistic Network Models*, Springer Verlag, New York, 1997.

[11] Friedman, N., and Koller, D., *Being Bayesian about Network structure*, UAI, 2000.

[12] Cooper, G.F., and Herskovits, E., *A Bayesian Method for the Induction of Probabilistic Networks from Data*, Machine Learning, 9:309-347, 1992.

[13] Heckerman, D., *A tutorial on lerning with Bayesian Network structures*, Lerning in Graphical Models, MIT Press, 1998.

[14] Dempster, A.P., Laird, N.M., and Rubin D.B., *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society, Series B,34:1-38, 1997.

[15] *Switchboard Corpora (LDC)*, http://www.ldc.upenn.edu/

[16] Thévenaz, P., *Résidu de Prédiction Linéaire et reconnaissance de locuteurs indépendante du texte*, Ph.D. Thèse. Université de Neuchâtel, Institut de Microtechnique, 1993.

[17] Faúndez-Zanuy, M., and Rodríguez-Porcheron, D., *Speaker Recognition Using Residual Signal of Linear and Nonlinear Prediction Models*, CICYT TIC97-1001-C02-02.

[18] Murphy, K.P., *The Bayes Net Toolbox for Matlab*, Computing Science and Statistics: Proceedings of the Interface, volume 33, 2001.

[19] Bilmes, J., and Zweig, G., *The Graphical Models Toolkit: An open source software system for speech and time-series processing*, Proc. IEEE ICASP 2002.

B.2 Workshop on Biometrics on the Internet. Vigo 2004

Eduardo Sánchez Soto, Raphaël Blouet, Marc Sigelle et Gérard Chollet.
Model Adaptation for Speaker Verification using Conditional Probability Tables.
Workshop on Biometrics on the Internet (COST 275).
Vigo (Espagne). Mars 25-26, 2004.

Model Adaptation for Speaker Verification using Conditional Probability Tables in Bayesian Networks

Eduardo Sánchez-Soto, Raphaël Blouet, Marc Sigelle and Gérard Chollet

École Nationale Supérieure des Télécommunications
Département de Traitement de Signal et des Images. LTCI/CNRS URA 820.
46 rue Barrault 75634 Paris Cedex 13 France
esanchez,blouet,sigelle,chollet@tsi.enst.fr

Abstract

In this paper a new adaptation technique for Speaker Verification of models built using Bayesian Networks is presented. The adaptation problem of parameters of the conditional probability tables (CPTs) is treated in a specific and new manner. A CPT transformation is made given that these tables are Stochastic Matrices. Model adaptation involves estimating the new vectors in the matrix with a transformation that includes vectors in the world model and the speaker model. The combination of both models is based on a value computed using a measure of distance between vectors of both CPTs. This speaker verification system has been tested using the NIST 2002 data base. Results show that this adaptation method improves the verification performances.

1. Introduction

Nowadays, an interest in Biometric Techniques for person identification is observed and Automatic Speaker Verification (ASV) is an important part of this growing field. This interest is justified by the persons wish of using their voice to reach several services where it is desirable to make their access safe. In the majority of real systems the quantity of available data is a problem. Complexity of data acquisition implies a problem for the creation of each speaker model. Therefore adaptation techniques are necessary and important in order to built models that perform better.

The objective of adaptation techniques for models creation is to adjust all the parameters of a specific model using new data. Several methods exist for models construction in the field of ASV and in consequence several methods for its adaptation. Some methods use statistical models like HMMs (Hidden Markov Models) [1] and GMMs (Gaussian Mixture Models) [2], where either the variances, or the means or both are adapted. Other techniques are DTW (Dynamic Time Warping)[3] or Neural Networks [4]. The first adaptation technique for ASV were proposed by Reynolds [5] and Mokbel in [6] has proposed a unified view. However, the models adaptation is always a very important problem in the systems of ASV. The suggested speech modeling with Bayesian Networks is not the exception. To solve this problem, a new technique of adaptation adapted to these types of Networks is proposed.

This article is organized as follows: in section 2, a short description of Bayesian Networks is made. Section 3 is dedicated to the adaptation methods and the presentation of our proposal using the Bayesian Networks. In section 4 experiments and results obtained are presented. Finally, conclusions and perspectives are given in section 5.

2. Bayesian Networks

In this section basic aspects of Bayesian Networks (BN), or Bayesian Belief Network [7], are presented. In the simplest form a BN is a probabilistic representation of the joint probability distribution defined on a finite set of random variables. This representation is based on probability theory and graph theory. A BN consists firstly of the structure given by a Directed Acyclic Graph (DAG).

The nodes in a DAG represent random variables and arcs represent conditional probabilistic dependencies among those variables. For example from the graph in Figure 1 an arc from A to C can be interpreted as indicating that A has a direct influence on C. Nodes have relative names by its position and relation with others nodes in the graph. Each edge points from one node, called parent, to another, called child, for example in the Figure 1 B is the parent of E. In the same topology description, the node X_j has a descendant node X_i if this one is its child or is connected to it through its children. In the same graph, E is a descendant of both A and B. In a BN, a Conditional Probability Distribution (CPD) is associated with each node X_i. It describes the dependency between this node and its parents. In general, each node is conditionally independent from its non-descendants given its parents. Those dependence relations induces a factorization in the joint distribution function expressed as :

$$P(X_1, \ldots, X_N) = \prod_{i=1}^{N} P(X_i | Pa(X_i)), \qquad (1)$$

where $Pa(X_i)$ is the set of X_i's parents. For the example in the graph in Figure 1 the factorization is :

$$P(A\,B\,C\,D\,E) = P(A)P(B)P(C|A) \\ P(D|A)P(E|A\,B). \qquad (2)$$

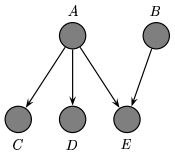


Figure 1: *A basic Bayesian Network example.*

Each Conditional Probability Distribution or Conditional Probability Tables represented by a factor in (1) describes the interaction between a node and its immediate predecessors. This representation form the second part of a BN. Specifically those CPTs represent the relation between parents and child. The child's value depends on the parents' values combination. For example, if all the variables are binary $X = \{x, \neg x\}$, the values $t_{i,j} = p(x = i|Pa(x) = j)$ in the CPT for the term $P(D|A)$ in (2) could be :

Table 1: CPT pour $P(D|A)$.

	a	$\neg a$
d	0.45	0.35
$\neg d$	0.55	0.65

2.1. Bayesian Network Construction

One main problem in probabilistic reasoning using Bayesian Networks is learning. Learning Bayesian Network from data consists in automatically building the network structure and compute the parameters, from information in data.

2.2. Structure Learning

Some aspects must be considered in the process of finding the best structure, even if the space of variables is fully observable. The amount of possible structures is very great. This quantity depend on the number of variables N, for example, if just four variables are taken into account 543 DAGs are possibles. It is unrealistic to explore all of them. For that reason, specific search algorithms must be designed.

Two different approaches to solve the structure learning problem exist. The first one searches in all the structure space and the second starts with a specific connected graph and then searches for independence relations in the data S, and puts in or takes away arcs. The initial graph can be place to put some prior knowledge about the variables relation.

The algorithm used in this work, K2 [8], belongs to the second approach. It uses a greedy search method to construct the structure. It starts with the the simplest structure, i.e. a graph without arcs. Given an ordering, or prior knowledge about relationship between the variables, the set of variables considered as candidate for the set of parents $Pa(X_i)$ for each variable X_i are restricted to those variables with smaller order numbers than X_i. All possible structures have to be scored to know which one has the highest quality. Maximum likelihood could be an adequate quality measure, but it privileges the fully connected graph, because it has the greatest number of parameters. Thus, a prior knowledge on the model can be used to overcome this problem. The marginal likelihood can be approximated [9] with a Laplace method, and finally get the BIC (Bayesian Information Criterion):

$$\log P(S|G) \approx \log P(S|G, \hat{\theta}) - \frac{d}{2} \log M, \qquad (3)$$

where S is the database, M is the number of samples, $\hat{\theta}$ is the estimate of the parameters and d is the dimension (number of free parameters) of the model.

2.3. Parameters Learning

It is required to adjust the parameters in CPDs in such a way that they describe the data statistically. The parameters θ and the model, $B(\theta)$, defined for these parameters are given. Also, the prior distribution over the models $P(B(\theta))$ and the space of parameters in these models $P(\theta|B)$ can be used. Therefore, given some data S, it is wanted to estimate θ, such that the posterior probability to be maximized is :

$$P(B|S) = \frac{P(B)}{P(S)} \int_{\theta} P(S|\theta, B) P(\theta|B) d\theta. \qquad (4)$$

Thus the maximum likelihood estimate of θ is computed by minimizing the cost function over the probability density function. We can make an optimization that relies on the gradient of this function, or use an iterative procedure like EM (Expectation - Maximization).

2.4. Bayesian Networks for Speaker Verification

Speech signal carries a lot of information besides the message. Other information about the speaker is present such as mood, emotive state and in particular his/her identity. All those informations, including spectral and prosodic, can be used in order to differentiate one speaker from others. BN could be the way to combine all off them [10, 11]. Conditional independencies can be retrieved directly from data in place to build a BN, and by this mean integrate those characteristics, in a probabilistic way.

3. Models Adaptation

In a ASV system a model called World model or Universal Background Model (M_m) is learned using a great quantity of data by hoping that the general characteristics of speakers can be well collected in the parameters of this model. This quantity of information is then adapted by using the data from each speaker (s_i). In this way one obtains each final speaker model (M_{s_i}). Then, models which depend directly on the initial world model and the new data D are produced :

$$M_{s_i} = F(M_m, D). \qquad (5)$$

Among the most used techniques are regression methods like MLLR (Maximum Likelihood Linear Regression) and the Bayesian estimate methods like MAP (Maximum A Posteriori).

3.1. Bayesian Network Adaptation

Knowing that BN are statistical models their parameters, especially CPTs, can be adapted by using regression or Bayesian methods. In this article we propose a CPTs adaptation of a BN based on the fact that each CPT is a stochastic matrix since verifies :

$$t_{i,j} \geqslant 0 \; \forall i,j \in B, \quad \sum_{i \in B} t_{i,j} = 1, \tag{6}$$

where B is the set of possible variable values by column. Each column of this matrix is a stochastic vector which under certain conditions is a good approximation of the probability density function (pdf).

Any modification to values in the pdf function implies necessarily modification to the dependencies between the variables modelled by RB. Then, this function can be used to perform adaptation. In this case the problem is brought back to a problem of comparison between two pdfs. On the basis of equation (5), CPTs of the final model will be a function of the CPT of the world model and the CPT obtained for the speaker before the adaptation:

$$CPT'_{s_i} = F(CPT_m, CPT_{s_i}), \tag{7}$$

where CPT'_{s_i} is the final model, CPT_m is the world model and CPT_{s_i} the model before adaptation. The adaptation using a combination of both initial CPTs is possible. A linear combination, that verifies the conditions in (6) is:

$$CPT'_{s_i} = \alpha CPT_m + (1-\alpha) CPT_{s_i}, \tag{8}$$

where $\alpha \in [0\,1]$. The value α could be a fixed value or can be obtained with a suitable distance computed between both *pdfs*. Two possible cases can be considered, a fixed α for all speakers or an α_i adapted for each speaker s_i.

4. Experiments and Results

In this section, experiments and results using our BN Speaker Verification System (BNSVS) are detailed.

4.1. Database

The data are taken from the second release of the Cellular Switchboard Corpus (Switchboard Cellular - Part 2) of the Linguistic Data Consortium. Each conversation is echo cancelled before use. The database is divided into training data (about 400 target speakers), and test data (about 600 test segments). The training data for a target speaker consist in about two minutes of speech from that speaker, excerpted from a single conversation. Actual duration is, however, constrained to lie within the range of 110 to 130 seconds. Each test segment is extracted from a 1 minute excerpt of a single conversation and is the concatenation of all speech from the subject speaker during the excerpt. The duration of the test segment therefore vary, depending on how much the segment speaker spoke. So, the effective speech duration lies between 15 and 45 seconds. Both test and target speakers are of the same sex.

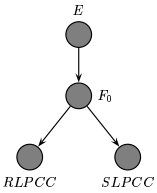


Figure 2: *Structure for the four variables (energy (E), pitch (F_0), signal SLPCC and residual RLPCC) issued from the K2 algorithm.*

4.2. Parameters

The training and test parameter vectors consist of a set of four types of parameters present all the 10 ms over a 20 ms window. The first vector is a 24-dimensional LP Cepstral Coefficients obtained as follow : 12-dimensional LPCC (Linear Prediction Cepstral Coefficients), with sliding CMS (Cepstral Mean Substraction) and augmented with their first derivatives, $SLPCC$, for Signal Linear Prediction Cepstral Coefficients. The second vector, 24-dimensional LPCC has been obtained as before from the LP-residual signal $RLPCC$ [12], and finally the frame pitch F_0 and the frame energy E as prosodic information are also used.

In a first test a discretization of all variables was achieved in order to simplify the probabilistic scheme. Variables $SLPCC$ and $RLPCC$ were discretized using 32 values, E with two and F_0 with three values (one value for unvoiced part). For the second set of tests the original variables were modeled by GMMs. $SLPCC$ and $RLPCC$ with 32 mixtures, F_0 with three mixtures and E with 2 mixtures.

4.3. Modeling

Parametrized data frames had been used with K2 algorithm and BIC score to find the best structure for the four variables. The structure obtained, Figure 2, is set to be speaker independent. Taking into account the relations of conditional independence issue from the graph in Figure 2 and equation (1) the joint probability distribution for the four variables becomes a product of local terms :

$$P(U) = P(E)P(F_0|E)P(RLPCC|F_0) \; P(SLPCC|F_0). \tag{9}$$

The relation between $SLPCC$, $RLPCC$ and F_0 is obtained from the term $P(RLPCC|F_0)P(SLPCC|F_0)$. It can be interpreted as a relation of conditional independence where $RLPC$ and $SLPC$ are independent given F_0. Also, it can be seen that F_0 depends directly on E.

4.4. Results

The decision score is directly based on the log-likelihood ratio between the target speaker and the world model over all the frames without any kind of normalization. Each test segment is evaluated against 11 hypothesized speakers. A first test was made using a fixed value α for all speakers and a discrete model. Then two different

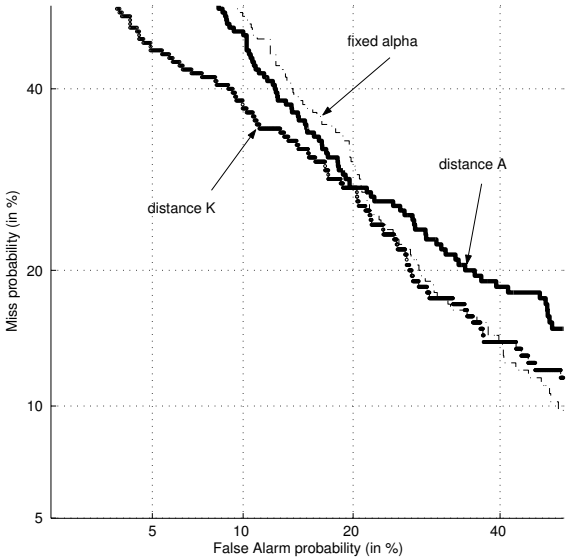


Figure 3: *DET Curves of the discret system using a fixed α and α obtained from Aitchinson distance d_A and Kullback-leibler simetric distance d_K.*

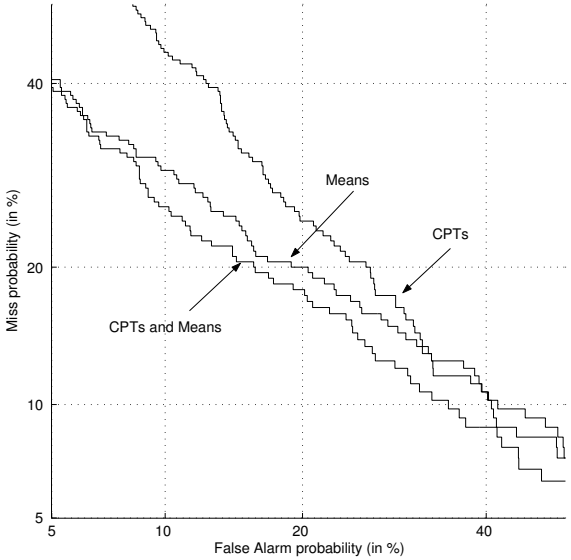


Figure 4: *DET curves for adaptation of different combination of parametres. First curve shows CPT adaptation with a fixed α. Second curve shows Bayesian means and the third Bayesian means and CPTs adaptations.*

distances were employed with the same discrete model. From (6), the $t_{i,j}$ values are subject to a unit sum constraint like proportions in a compositional data. Then, in Aitchinson geometric [13] structure of probability functions on finite intervals (a, b), a distance can be defined for any two *pdf*, f and g like:

$$d_A(f,g) = \left[\frac{1}{4L}\sum_{x=a}^{b}\sum_{y=a}^{b}\left(\log\frac{f(x)}{f(y)} - \log\frac{g(x)}{g(y)}\right)^2\right]^{1/2}. \tag{10}$$

where $L \in [a, b]$ is the interval's length. The second used distance was the Kullback-leibler simetric distance :

$$d_K(f,g) = \sum_{x=a}^{b} f(x)\log\frac{f(x)}{g(y)}. \tag{11}$$

The performance of the system with these conditions is shown in the DET plot in Figure 3, where it can be seen the influence of α. The Kullback-leibler distance shows the best performance.

In order to validate our system a comparison with a classic Bayesian GMM adaptation techniques was made in the second set of experiments. A single adaptation coefficient for all means was used. Results in Figure 4 are obtained using the GMM modelisation for the four continuous variables and CPTs for its relations. The performance of three systems are compared in Figure 4. The first system use a CPTs adaptation, the second a mean adaptation and the third system shows the scores ontained employing means and CPTs adaptation.

From this DET curve is concluded that adaptation of conditional independence relations in CPTs joined to means adaptation perform better that just a mean or CPT adaptation.

5. Conclusions

In this paper, a new adaptation technique for Speaker Verification based on BNs is presented. CPTs are intuitively viewed as probability density functions. A distance measure between two different CPTs is used to compute a value that controls the combination of both CPTs. The obtained results show that BNs are a flexible mathematical tool that can help to modelize information from different aspects of the speech signal and also suggest several research lines concerning the used distance. The perspectives for this work are important because of the flexibility of BNs.

6. References

[1] Matsui T. and Furui S., "Speaker adaptation of tied-mixture-based phoneme models for text-prompted speaker recognition", Proceedings ICASSP, 1125-1128, 1994.

[2] Gauvain J. L. and Lee C. H., "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains", "IEEE Trans. Speech Audio Processing, 2:291-298, 1994.

[3] Naik J. M. and Doddington G. R., "High performance speaker verification using principal spectral components", Proceedings ICASSP, 881-884, 1986.

[4] Mistretta W. J. and Farrell K. R., "Model adaptation methods for speaker verification", Proceedings ICASSP, 1998.

[5] Reynolds D. A., "Comparison of background normalization methods for text independent speaker verification", Eurospeech, 1997.

[6] Mokbel C., "Online adaptation of hmms to real life conditions: A unified framework", IEEE Trans. on Speech and Audio Processing, Vol. 9, No. 4, May 2001.

[7] Perl J., "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference", Morgan Kaufmann, San Diego, 1988.

[8] Cooper G.F. and Herskovits E., "A Bayesian Method for the Induction of Probabilistic Networks from Data", Machine Learning, 9:309-347, 1992.

[9] Heckerman D., "A tutorial on learning with Bayesian Network structures", Lerning in Graphical Models, MIT Press, 1998.

[10] Sanchez-Soto E., Blouët R., Chollet G. and Sigelle M., "Speaker Verification with Bayesian Networks", Workshop on Multimodal User Authentication", pp. 61-65, Santa Barbara, CA. Dec. 11-12, 2003.

[11] Arcienega M., Drygajlo A., "A Bayesian Network Approach for Combining Pitch and Spectral Envelope Features for Speaker Verification", COST 275 Workshop "The Advent of Biometrics on the Internet", pp. 99-102, Rome, Italy, Nov. 7-8, 2002.

[12] Thévenaz P., "Résidu de Prédiction Linéaire et reconnaissance de locuteurs indépendante du texte", Ph.D. Thèse. Université de Neuchâtel, Institut de Microtechnique, 1993.

[13] Aitchison J., "The Statistical Analysis of Compositional Data", Chapman and Hall Ltd., London (UK), 1986.

B.3 Odyssey-04 The ISCA Speaker and Language Recognition Workshop. Toledo 2004

Raphaël Blouet, C. Mokbel, H. Mokbel, E. Sánchez Soto, Gérard Chollet et H. Greige.
BECARS: a free software for speaker verification
Odyssey-04 The ISCA Speaker and Language Recognition Workshop.
Toledo (Espagne). Mai 31-Juin 3, 2004.

BECARS : a free software for speaker verification

Raphaël Blouet†, Chafic Mokbel‡, Hoda Mokbel‡, Eduardo Sánchez Soto†, Gérard Chollet† and Hanna Greige‡

†ENST, dépt. TSI
46 rue Barrault, 75634 Paris
France
{blouet,esanchez,chollet}@tsi.enst.fr

‡University of Balamand
El-Koura,BP 100 Tripoli
Lebanon
{chafic.mokbel,hanna.greige}@balamand.edu.lb

Abstract

The aim of Automatic Speaker Verification (ASV) is to detect whether a speech segment has been uttered by the claimed identity or by an impostor. Our contribution includes the distribution of BECARS , a free software based on Gaussian Mixture Models (GMM) for Automatic Speaker Verification (ASV), and the design of a new methodology to estimate the decision score in an ASV system. BECARS in available at http://www.tsi.enst.fr/~blouet/Becars/. The main characteristic of this software is to allow the use of several adaptation techniques including the most common ones such as *Maximum A Posteriori* (MAP) and *Maximum Likelihood Linear Regression* (MLLR). The proposed method for score computation is based on the use of a hierarchical Gaussian clusterization method that we describe in details in this paper.
We introduce this work with a general summary of Automatic Speaker Verification (ASV), followed by a description of the adaptation technique available in BECARS used in this work. We then present and evaluate our score computation scheme before concluding the paper.

1. Introduction

Given a speech segment $\underline{\underline{Y}} = \{\underline{Y}_1, \cdots, \underline{Y}_N\}$ and a hypothesized (or claimed) identity X, the aim of Automatic Speaker Verification is to determine whether $\underline{\underline{Y}}$ has been uttered by X or not.

Automatic Speaker Verification is often formulated as a classical hypothesis test between two hypotheses H_X and $H_{\bar{X}}$ with:

H_X : $\underline{\underline{Y}}$ has been uttered by X
$H_{\bar{X}}$: $\underline{\underline{Y}}$ has been uttered by another speaker

The optimal test to decide between these two hypotheses is the likelihood ratio test:

$$S_X(\underline{\underline{Y}}) = \log \frac{p_X(\underline{\underline{Y}})}{p_{\bar{X}}(\underline{\underline{Y}})} \underset{H_{\bar{X}}}{\overset{H_X}{\gtrless}} \theta \qquad (1)$$

where θ is the decision threshold.

This approach relies on the hypothesis of the existence of both probability density functions p_X and $p_{\bar{X}}$ on the whole observation space $\mathbf{Y}$ of frames $\underline{Y}_t$.

For a decade, the state of the art approach consists in using Gaussian Mixture Models [9] to modelize both probability density functions. Moreover, training of each client model is mostly done by adaptation of $p_{\bar{X}}$ parameters [10]. BECARS allows the use of several kinds of adaptation criterions. In the next section, we describe the one that we used for this work. More details on adaptation techniques available in BECARS can be found in [2] or in the software documentation.

2. Adaptation techniques for client models determination

Hidden Markov Model (HMM) and GMM adaptation techniques have largely been studied in the last decade. In [6] a unified theoretical framework has been proposed in which the two major classes of techniques, *Maximum A Posteriori* (MAP) and transformation based adaptation (like *Maximum Likelihood Linear Regression* (MLLR)), appear as particular cases. Several adaptation techniques have been derived within this framework and applied in BECARS in order to determine client models.

Model adaptation can be seen as a particular case of training where a small amount of uncontrolled data is used to estimate new values for the parameters. In such cases, the training must be controlled in order to ensure that the estimated parameters are not specific to the training data. In order to incorporate this idea, adaptation is seen as a function with a variable degree of freedom that transforms the values of the parameters. The degree of freedom must be chosen as a function of the available training data. In order to achieve this variable degree of freedom, the general adaptation theory proposed in [6] matches a binary tree with a GMM. As shown in Figure 1 (with a 4 Gaussian components GMM), each component of the GMM stands for one leaf of the tree. From the root of the tree to the leaves, different cuts can be defined allowing different possible distribution classifications. For every possible classification, a number of classes is obtained. In each of these classes, an adaptation function

may be associated.

In order to build the tree, Gaussian distributions are grouped two by two up to the root. At each grouping step, we chose the two closest distributions given the distance $d(\cdot,\cdot)$. The distance $d(\mathcal{N}_1,\mathcal{N}_2)$ between two Gaussian $\mathcal{N}_1$ and $\mathcal{N}_2$ is defined as the likelihood loss on the training frames if $\mathcal{N}_1$ and $\mathcal{N}_2$ are replaced by a single distribution $\mathcal{N}_3$. In the tree construction, $\mathcal{N}_3$ is associated with the $\mathcal{N}_1$ and $\mathcal{N}_2$ parent node.

$$d(\mathcal{N}_1,\mathcal{N}_2) = \log \frac{|\underline{\underline{\Sigma_3}}|^{\frac{\alpha_1+\alpha_2}{2}}}{|\underline{\underline{\Sigma_1}}|^{\frac{\alpha_1}{2}} |\underline{\underline{\Sigma_2}}|^{\frac{\alpha_2}{2}}} \quad (2)$$

On equation 2, $|\underline{\underline{\Sigma_1}}|$ and $|\underline{\underline{\Sigma_2}}|$ are the covariance matrices of the two nodes and $|\underline{\underline{\Sigma_3}}|$ is the covariance matrix of an equivalent node. α_1 and α_2 are the training factors for the two nodes respectively.

In order to perform adaptation, the standard **EM** algorithm [3] is modified.

At the end of the **E** step, the weights associated with Gaussian distributions forming the leaves of the tree are propragated up to its root. Then, starting from the root, the tree is processed and we stop at nodes whose children have weights less than a predefined threshold. This predefined threshold represents the minimal amount of data necessary to perform the adaptation. This allows the determination of a classification that is a function of the amount of available data. This process is described in the Figure 1.

At the **M** step of the **EM** algorithm, a regression function is associated with every class of Gaussian distributions and every dimension of the acoustic feature space. All the Gaussian distributions in the class will have their mean and variance adapted as following:

$$\begin{aligned} \mu_i &= a_i m_i + b_i \\ \sigma_i^2 &= a_i^2 s_i^2 \end{aligned} \quad (3)$$

with:

- m_i and s_i^2 respectively the mean and the variance of the i^{th} *prior* distribution of the *prior* GMM,
- μ_i and σ_i^2 respectively the mean and variance of the adapted Gaussian distribution,
- a_i and b_i, parameters of the regression function.

General equations for the estimation of regression parameters in the framework of the unified adaptation theory are given in [6]. Here, we only consider the case of one particular adaptation called MLLR_MAP in BECARS . Equations 4 and 5 allow us to obtain the regression coefficients and equation 6 presents reestimation formulae obtained in this case. In equations

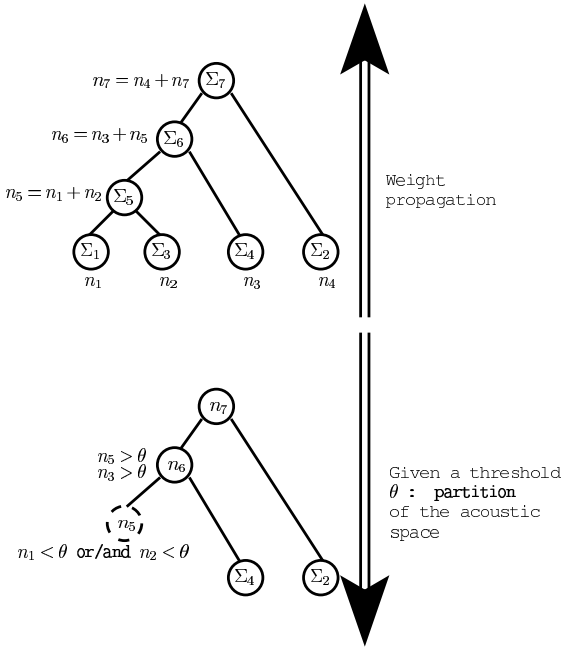


Figure 1: Description of the two steps that permit the determination of the number of degrees of freedom used for adaptation

4 and 5, J is the number of Gaussian distributions in the q^{th} class, n_j is the weight of the j^{th} distribution after the **E** step, $r0_j$ is the *a priori* precision, m_j the *a priori* mean, $\bar{y}_j$ and $\bar{y}_j^2$ are the first and second order moments observed after the **E** step.

$$\begin{aligned} 0 &= |a_q^2| \cdot \left[\sum_{j=1}^{J} n_j\right] + |a_q| \cdot \left[\sum_{j=1}^{J} r0_j \cdot n_j \cdot m_j \cdot \bar{y}_j \right. \\ &\quad \left. - \frac{\left(\sum_{j=1}^{J} r0_j \cdot n_j \cdot \bar{y}_j\right) \cdot \left(\sum_{k=1}^{J} r0_k \cdot n_k \cdot m_k\right)}{\sum_{j=1}^{J} r0_j \cdot n_j}\right] \\ &\quad - \left[\sum_{j=1}^{J} r0_j \cdot n_j \cdot \bar{y^2}_j - \frac{\left(\sum_{j=1}^{J} r0_j \cdot n_j \cdot \bar{y}_j\right)^2}{\sum_{j=1}^{J} r0_j \cdot n_j}\right] \end{aligned} \quad (4)$$

$$b_q = \frac{\sum_{j=1}^{J} [r0_j \cdot n_j \cdot (\bar{y}_j - a_q \cdot m_j)]}{\sum_{j=1}^{J} [r0_j \cdot n_j]} \quad (5)$$

Equation 4 shows that the regression coefficient a_q is the solution of the second degree equation. As shown in [6], this equation always has two solutions of opposite sign. In order to smooth further the adaptation, an empiric *Maximum A Posteriori* (MAP) is applied to the

estimation of the mean :

$$\mu_i = 0.8 \cdot (a_i m_i + b_i) + 0.2 \cdot m_i \quad (6)$$

3. Frame weighting using MMI

GMM modeling of speech frames represented by their corresponding feature vectors does not take into consideration the order of the frames. This means that rearranging the speech signal frames will not affect their likelihood computed using the GMM. Thus, the collection of frames from a speaker's utterance available in a test represents a sample from the speaker population of frames. Moreover, the GMM-based Automatic Speaker Verification system can be viewed as calculating the expected value over this sample of a decision function. In our case, this decision function corresponds to the log likelihood ratio betwenn hypothesis H_X and $H_{\bar{X}}$.

Let $\underline{Y}_t$ be the feature vector representing a speech frame extracted from a test speech utterance; $\underline{Y}_t$ is supposed to be the realization of a random multivariate variable $\underline{Y}$. Let $llr(\underline{Y})$ be the log likelihood ratio function considered as the main argument of the decision function; a theoretical expected value of $llr(\underline{Y})$ is given by:

$$\begin{aligned} LLR &= E[llr(\underline{Y})] \\ &= \int_{\Omega_Y} llr(\underline{Y}) p(\underline{Y}) d\underline{Y} \end{aligned} \quad (7)$$

If we assume that the process underlying the production of $\underline{Y}$ is ergodic, it is equivalent to estimating $llr(\underline{Y})$ over the parameter space than estimating it with a time average. This explains why the score for a given utterance of T frames is calculated as:

$$\hat{LLR} = \sum_{t=1}^{N} llr(\underline{Y}_t) \quad (8)$$

However, a signal frame carries different information about the underlying speech message such as the speaker's identity, prosody, the communication channel, etc. Let us define the binary random variable I_S representing the fact that a signal frame carries information about the speaker identity. To illustrate this idea we simply cite the obvious example of silence signal frames which, in general, carry little information about the speaker identity. Using this random variable, a better estimation of the LLR may be derived from 7:

$$\begin{aligned} LLR &= E_{I_s}[llr(\underline{Y})] \\ &= \int_{\Omega_Y} llr(\underline{Y}) p(I_s|\underline{Y}) P(\underline{Y}) d\underline{Y} \\ &\propto \int_{\Omega_Y} llr(\underline{Y}) p(\underline{Y}|I_s) d\underline{Y} \end{aligned} \quad (9)$$

Therefore, and even if the process of generating the feature vectors is supposed to be ergodic, the average in equation 9 is not equivalent to an average over time. The choice of the sample for estimating the LLR value should be done with care. An alternative consists in weighting the instantaneous LLR measurement by a factor that depends on the relevance of the corresponding frame regarding the characterization the speaker identity.

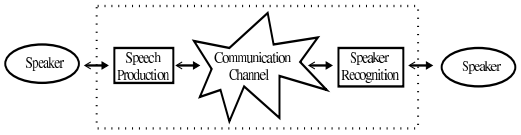


Figure 2: *View of the global chain of a speaker recognition system.*

Several approaches exist in order to calculate these weights. In the present work, we propose to perform a vector quantization and to associate a weight with each feature subspace defining a class. If $C(\underline{Y})$ defines the class of a feature vector $\underline{Y}$, we approximate $p(I_s|\underline{Y})$ in the equation 9 by a discrete distribution defined by $p(I_s|C(\underline{Y}))$.

In this paper and in order to determine parameters of the discrete probability distribution $\{p(I_s|C(\underline{Y}))\}$, the maximization of the Mean Mutual Information (MMI) is used. To illustrate the principle of using the MMI criterion, the complete chain of an Automatic Speaker Recognition system is provided in Figure 2. A given speaker utters a speech signal which goes across a communication channel to reach the ASV system that is used to determine the identity of the speaker. In this model, we suppose that a communication channel is defined going from the speech production module to the speaker recognition module included. In the development phase, we add the true speaker identity to the input of the ASV system and we optimize $\{p(I_s|C(\underline{Y}))\}$ to have the output identity as close as possible to the one provided in the input. To summarize, we want to obtain a maximum of the information that has been put into the communication channel or to maximizes the Mean Mutual Information. In the develpement phase of our ASV system, we chose the weights of the discrete probability distribution $\{p(I_s|C(\underline{Y}))\}$ that maximise this information. Figure 3 summarises the process of the weights estimation.

4. Evaluation protocol and result

Evaluations related in this section are made using the NIST 2003 data set [8].

Acoustic parametrisation consists in 20 mel cepstra filter bank coefficients with their delta. Channel equalisation is performed through Feature Warping [7].

We use a 128 components GMM to model p_X and $p_{\bar{X}}$. For each speaker p_X is obtained by adaptation of $p_{\bar{X}}$ following equation 6. $p_{\bar{X}}$ is trained using 2 hours

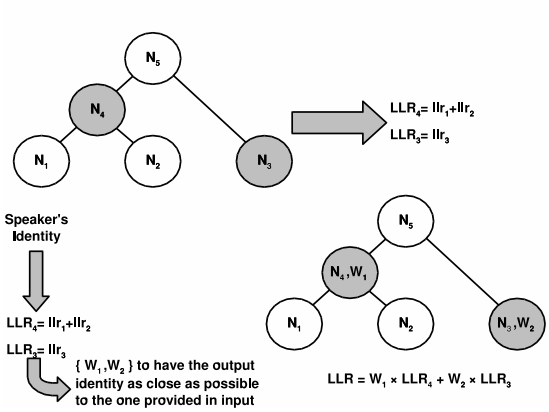


Figure 3: *Description of the MMI based score computation strategy.*

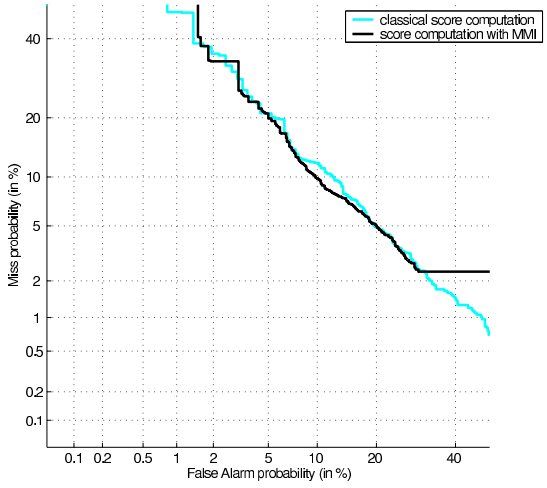


Figure 4: *DET curves associated to the use or not of the MMI criterion.*

of speech uttered from 100 speakers of the NIST 1999 evaluation campaign. The discrete probability distribution $\{p(I_s|C(\underline{Y}))\}$ is estimated using 500 test files and 50 speakers of the NIST 2003 evaluation data. In the results presented here, we used 32 differents classes. The two Det curves [5] obtained with and whithout the use of the frame weighting are plotted in Figure 4. Unless score computation using the MMI approach performs slightly better than the classic score computation strategy, both systems have very close level of performance. We believe that this can be explained mostly by the lack of data available to estimate the discrete probability distribution $\{p(I_s|C(\underline{Y}))\}$.

5. Discussion and conclusion

As the proposed approach appears theoretically very promising, results obtained on the NIST 2003 data set are not as good as expected. However, we still believe that this approach may improve ASV system robustness and we will run further experiments using differents configurations. The use of several strategies and criterion to estimate the discrete probability distribution $\{p(I_s|C(\underline{Y}))\}$ will also be investigated.

6. References

[1] R. Auckenthaler, M. Carey and H. Lloyd-Thomas, *Score Normalization for Text-Independant Speaker Verification Systems*, Digital Signal Processing Vol 10., Nos 1-3, Janvier 2000.

[2] R. Blouet, C. Mokbel and Gérard Chollet, BECARS *: un logiciel libre pour la vrification du locuteur*, JEP 2004, Fèz, April 2004.

[3] A.P. Dempster, N.M. Laird, D.B. Rubin, *Maximum likelihood from incomplete data using the EM algorithm*, Journal of the Royal Statistical Society, 39(B), 1977.

[4] Y. Linde, A. Buzo et R. Gray,*An Algorithm for Vector Quantizer Design*, IEEE Transactions on Communications,1980.

[5] A. Martin, G. Doddington, T. Kamm, M. Ordowski and M. Przybocki, *The DET Curve in Assessment of Detection Task Performance*, Proceedings of EuroSpeech 1997, Volume 4, pp. 1895-1898.

[6] C. Mokbel, *Online adaptation of hmms to real life conditions: A unified framework*, IEEE Transaction on Speech and Audio Processing, 2001.

[7] J. Pelecanos and S. Sridharan, *Feature Warping for Robust Speaker Verification*, Workshop Odyssey, 2001.

[8] M. Przybocki and A. Martin, *The NIST Year 2003 Speaker Recognition Evaluation Plan*, 2003.

[9] D.A. Reynolds, *A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification*, Georgia Institute of Technology, 1992.

[10] D.A. Reynolds, *Comparison of background normalization methods for text independent speaker verification*, Eurospeech'97, 1997.

Acknowledgement:
This work has partly been funded by CEDRE, a French-Lebanese cooperation framework (involving ENST and UOB). It has also benefited from the scientific support of the ELISA consortium.

B.4 International School on Nonlinear Speech Processing. Vietri Sul Mare 2004

E. Sánchez-Soto, M. Sigelle and G. Chollet.
Graphical Models for Text Independent Speaker Verification
International School on Nonlinear Speech Processing. Vietri Sul Mare, Italie. Septembre 13-18 2004.
Selected and published by Springer in Lecture Note in Computer Science Series (LNCS), vol. 3445

Graphical Models for Text-Independent Speaker Verification

Eduardo Sánchez-Soto, Marc Sigelle, and Gérard Chollet

École Nationale Supérieure des Télécommunications,
Département de Traitement du Signal et des Images, CNRS UMR LTCI
46, rue Barrault 75634 Paris Cedex 13 FRANCE
esanchez,sigelle,chollet@tsi.enst.fr,
http://www.tsi.enst.fr/~esanchez

Abstract. Our approach in text independent Speaker Verification (SV) proposes to integrate different aspects of the speech signal which convey information about the speaker's identity using Graphical Models (GM). Prosodic, spectral and source information obtained from the residue of linear prediction analysis are modeled in a probabilistic framework with a system based on Bayesian Networks (BN). The structure, or conditional independencies between the variables, is learned directly from the data using two different algorithms. In particular, the interpretation and comparation of the structures is presented. Some experiments conducted on the NIST 2003 one speaker text-independent data base have been conducted to demonstrate the feasibility of this approach.

1 Introduction

The performance of speech processing systems in some cases is still far from that of humans. At the decision step, a difference between those systems and humans is in one hand the used information quantity and in the other hand the relationships made between those informations. The spectral and prosodic aspects of speech signal are an abundant source of knowledge, but a joint representation of those aspects is until now a problem. The state-of-the-art SV systems use Gaussian Mixture Models (GMM) [7] in order to represent the data distribution. All the data are represented in a single space where no difference is made between the data that comes from one source or another. To overcome this problem GM can be used. GM are naturally modular and can represent in a visual way the relations between different variables in a given problem. Particularly, a BN [6] is a GM which represents in an optimal way conditional independencies between a set of random variables. Then, various aspects of speech signal can be jointly represented in a formal mathematical way using different variables which are related in a BN.

The relationship among the variables can be defined by an expert using some knowledge about the variables or by a learning technique that is applied directly to the data. The first work done in automatic learning of the structure in a BN [1] was able to obtain a simple tree structure from a database. Later an alternate

approach [2] which works also with multiply-connected networks was proposed. This technique is based on a Bayesian approach which assumes a prior uniform distribution over all the structures. Another approach is based on the principle of Minimal Description Length (MDL). From the Bayesian point of view the MDL approach assumes a prior distribution over the models which is inversely proportional to their encoding length.

However, finding the best structure, the conditional independencies which best represent the present relationships into the database, is a research field very important. Then we propose a technique to score the structure based on the MDL principle and its comparation with another structure obtained using other quality measure.

The organization of the paper is as follows. In section 2, we will first introduce Bayesian Networks. Section 3 reviews briefly some ideas about structure learning in BNs specially the proposed MDL technique, section 3.2. In section 4, we will present the experiments, results and its probabilistic interpretation. Finally we will give our conclusions in section 5.

2 Bayesian Networks

A BN [6] makes a representation of a joint probability distribution defined on a finite set of random variables. The nodes in a Directed Acyclic Graph (DAG) represent random variables and arcs represent conditional probabilistic dependencies among those variables. Nodes have relative names by its position and relation with others nodes in the graph. Each edge points from one node, called parent, to another, called child.

In a BN, a Conditional Probability Distribution (CPD) is associated with each node X_i. It describes the dependency between this node and its parents. In general, each node is conditionally independent from its non-descendants given its parents. Those dependence relations induces a factorization in the joint distribution function expressed as :

$$P(X_1, \ldots, X_N) = \prod_{i=1}^{N} P(X_i | Pa(X_i)), \tag{1}$$

where $Pa(X_i)$ is the set of X_i's parents.

Each Conditional Probability Distribution or Conditional Probability Tables represented by a factor in (1) describes the interaction between a node and its immediate predecessors.

3 Structure Learning

Learning Bayesian Network from data consists in automatically building the network structure and compute the parameters, from information in data. Some aspects must be considered in the process of finding the best structure, even if

the space of variables is fully observable. The amount of possible structures is very large. This quantity depend on the number of variables in an exponential way. To manage this problem two different approaches exists. One searches in all the structure space and the other starts with a specific connected graph which can be place to put some prior knowledge about the variables relation. Then it searches for independence relations in the data S, putting in or taking away arcs.

The algorithm used in this work, K2 [2], belongs to the second approach. It uses a greedy search method to construct the structure. It starts with the simplest structure, i.e. a graph without arcs. Given an ordering, or prior knowledge about relationship between the variables, the nodes considered as candidate for the set of parents $Pa(X_i)$ for each variable X_i are restricted to those nodes with smaller order numbers than X_i. All possible structures have to be scored to know which one has the highest quality.

3.1 Bayesian Information Criterion

Since a fully connected graph has the greatest number of parameters the maximum likelihood is not an adequate quality measure. Thus, a prior knowledge on the model can be used to overcome this problem. Let G be the structure and S the database or sequence of N samples for all the nodes of G. The posterior probability of data is :

$$P(G|S) \propto P(S|G)P(G), \tag{2}$$

where $P(G)$ is the prior probability of structure. Now, the likelihood of data obtained by integration on the possible values of parameters θ is :

$$P(S|G) = \int P(S|G,\theta)P(\theta|G)d\theta. \tag{3}$$

The marginal likelihood can be approximated [4] with a Laplace method, and finally get the Bayesian Information Criterion (BIC) :

$$\log P(S|G) \approx \log P(S|G,\hat{\theta}) - \frac{d}{2}\log M, \tag{4}$$

where M is the number of samples, $\hat{\theta}$ is the estimate of the parameters and d is the dimension (number of free parameters) of the model.

3.2 MDL

MDL [8] is used for the encoding of the data given a model. From equation (3) and from its limited expansion up to the second order one has :

$$P(S|G) \approx L(\hat{\theta}) \int exp - \frac{1}{2}(\theta-\hat{\theta})^t \, \mathbf{A}(\theta-\hat{\theta})d\theta \approx L(\hat{\theta})(\frac{1}{2\pi})^{\frac{d(G)}{2}} \frac{1}{\sqrt{det\ \mathbf{A}}}, \tag{5}$$

where $d(G)$ is the number of parameters specifying the model G. At the lowest order if N is the number of observations the MDL equation is obtained :

$$\mathcal{L} = \log P(S|G) \approx L(\hat{\theta}) - \frac{d(G)}{2} \log N. \tag{6}$$

Now, for a tree structure and in particular when all the nodes are independents it can be written :

$$\log P(S|G) \approx \sum_{s \in G} \{ [\sum_{i \in \Omega^s} (N_i^s) + \frac{1}{2} \log N_i^s] - (N + \Omega^s| - \frac{1}{2}) \log N + (|\Omega^s| - 1) \log \sqrt{2\pi} \}, \tag{7}$$

where Ω^s is the set of observable states at generic node s, and N_i^s is the observed number of times of the variable s in the state i.

3.3 Modelisation

The training and test parameter vectors consist of a set of four types of parameters. The first vector is a 24-dimensional LP Cepstral Coefficients obtained as follow : 12-dimensional LPCC, with sliding CMS and augmented with their first derivatives, $SLPCC$, for Signal Linear Prediction Cepstral Coefficients. The second vector, 24-dimensional LP Cepstral Coefficients has been obtained as before from the LP-residual signal $RLPCC$, and finally the frame pitch F_0 and the frame energy E. A gender-dependent Universal Background Models (UBM) have been created using part of the 2001 cellular development and evaluation datasets (similar to the database described in section 4).

First, those data had been used with K2 algorithm to find the best structure for the four variables. It has worked with all the possible orders and used the BIC score [4]. From this analysis we have obtained the conditional independence relations that define the first network structure which is set to be speaker independent, Figure 1.

Fig. 1. *Structure for the four variables (energy, pitch, signal lpcc and residual lpcc) issue of the K2 algorithm with BIC (left) and issue of the MDL analysis (right).*

From this graph we can write the joint probability distribution for these four variables $U = \{E, F_0, RLPCC, SLPCC\}$ as :

$$P(U) = P(E)\ P(F_0|E)\ P(RLPCC|F_0)\ P(SLPCC|F_0). \quad (8)$$

The terms $P(RLPC|F_0)P(SLPC|F_0)$ can be interpreted as $I(RLPC \perp SLPC|F_0)$, that is, the $RLPC$ and $SLPC$ are independent given F_0. The second term $P(F_0|E)$ reflects the close relation between energy and voicing in speech.

In a second part the MDL algorithm was performed using discrete data in order to simplify the probabilistic scheme. Those data were obtained using Vector Quantization (VQ) of variables initialized with the k-means algorithm. $SLPCC$ and $RLPCC$ variables were discretized using 32 values, E with two values and F_0 with three values (one value corresponding to the unvoiced parts). The obtained structure is shown in Figure 1. The conditional probability density for the four variables given the structure is :

$$P(U) = P(SLPCC)\ P(E|SLPCC)\ P(F_0|SLPCC)\ P(E|SLPCC). \quad (9)$$

Thinking about the SLPCC coefficients computation it is easy to see that those coefficients contain a lot of information which depend on the p number used in the autocorrelation function computation. In these coefficients one can find the excitation, the energy and then also the pitch characteristics since the LP model is not perfect.

Once the structure has been learned, the final world model uses a Gaussian Mixture (GM) implemented with BN to represent each variable (32 Gaussians for RLPCC and SLPCC, five for the pitch and two for the energy). The parameters were then learned with EM [3]. LBG algorithm was used to determine the initial setting. Target Speaker Models have been obtained by adaptation of the means in the world model by three iterations of the EM algorithm initialized with the world model.

4 Experiments and Results

The data are taken from the second release of the Cellular Switchboard Corpus of the Linguistic Data Consortium (LDC) [5]. The experiments were done using a half part of the male test database, 751 files. Each test file is tested against 11 speaker models. Then there are 8261 tests in total.

The decision score is directly based on the log-likelihood ratio between the target speaker and the world model over all the frames without any kind of normalization. The results in the Figure 2 show the influence of the structure in the final results. The structure obtained with MDL perform better than the structure obtained with K2 and BIC if the arcs between the continuous variables that model the mixture of gaussians are used. With the discrete relations the best performance is obtained with the K2 and BIC structure. The used relations to relate the variables (discrete or continuous) does not affect to much to the MDL structure, but it is not the case for the other structure which change in more than 2%.

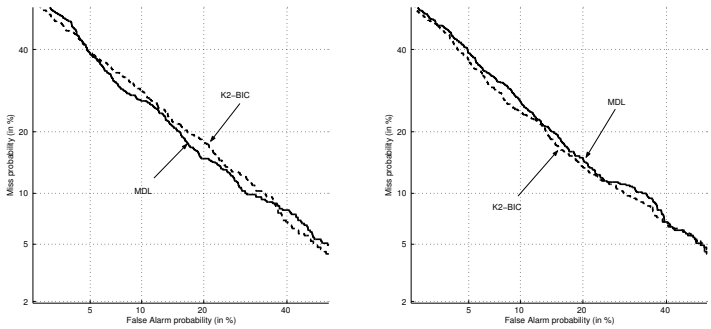


Fig. 2. *Results obtained using the continuous relation (left) and discrete relation (right).*

5 Conclusions

In this paper, a system achieving SV based on BNs is presented. This system infers the BN structure automatically from the data using two different quality measure functions. The obtained structures are compared and used for integrate all the information presented on the speech signal in a single probability distribution. Results reflect the influence of conditional independencies used in each model. It also shows that BNs are a flexible mathematical tool that can help to model information from different aspects of the speech signal. The physical interpretation given to the equations describing the structures suggests that the learning algorithms for BN are able to adequately infer the relations present in data.

References

1. C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.
2. G. F. Cooper and E. Herskovits. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning*, 9:309–347, 1992.
3. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 34:1–38, 1997.
4. D. Heckerman. *A tutorial on Learning with Bayesian Network Structures. Lerning in Graphical Models.* MIT Press, Cambridge, 1998.
5. Switchboard Corpora LDC. http://www.ldc.upenn.edu/.
6. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Diego, 1988.
7. D. A. Reynolds. *A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification.* PhD thesis, Georgia Institute of Technology, 1992.
8. J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

Bibliography

Acid, S., de Campos, L. M., Fernández-Luna, J. M., Rodríguez, S., Rodríguez, J. M., and Salcedo, J. L. (2004). A comparison of learning algorithms for Bayesian networks: a case study based on data from an emergency medical service. *Artificial Intelligence in Medicine*, **30**, 215–232.

Aitchison, J. (1986). *The Statistical Analysis of Compositional Data*. Chapman and Hall Ltd London (UK).

Alizé: a free, open tool for speaker recognition (2004). http://www.lia.univ-avignon.fr/heberges/alize.

Andersen, M. N., Andersen, R., and Wheeler, K. (2004). Filtering in Hybrid Dynamic Bayesian Networks. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume V, pages 773–776.

Arcienega, M. and Drygajlo, A. (2002). A Bayesian Network Approach for Combining Pitch and Spectral Envelope Features For Speaker Verification. In *COST 275 Workshop - The Advet of Biometrics on the Internet*, pages 99–102.

Auckenthaler, R., Carey, M., and Lloyd-Thomas, H. (2000). Score Normalization for Text-Independent Speaker Verification System. *Digital Signal Processing*, **1**.

Bartlett, M. S. (1935). Contingency Table Interactions. *Journal of the Royal Statistical Society, Supplement*, **2**, 248–52.

Ben, M., Blouet, R., and Bimbot, F. (2002). A Monte-Carlo Method for Score Normalization in Automatic Speaker Verification using Kellback-Leiber Distance. *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP'02)*, **1**, 689–692.

Bilmes, J. and Zweig, G. (2002). The Graphical Models Toolkit: An Open Source Software System for Speech and Time-series Processing. *IEEE ICASP*.

Bilmes, J., Zweig, G., Richardson, T., Filali, K., Lverscu, K., Xu, P., Jackson, K., Brandman, Y., Sandness, E., Holtz, E., Torres, J., and Byrnes, B. (2001). Discriminatively Structured Dynamic Graphical Models for Speech Recognition. In *JHU Summer Workshop*.

Bilmes, J. A. (1999). Buried Markov Models for Speech Recognition. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 713–716.

Bilmes, J. A. (2003). Buried Markov Models : A graphical-modeling approach to automatic speech recognition. *Computer Speech and Language*, **17**, 213–231.

Bimbot, F., Bonastre, J. B., Fredouille, C., Gravier, G., Magrain-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-García, J., Petrovska-Delacrétaz, D., and Reynolds, D. A. (2004). A Tutorial on Text-Independent Speaker Verification. *EURASIP Jouranl on Applied Signal Processing*, **4**, 430–451.

Blalock, H. M. (1971). *Causal Models in the Social Sciences*. Aldine-Atheston, Chicago.

Blouet, R., Mokbel, C., Sánchez-Soto, E., Chollet, G., and Greige, H. (2004). BECARS: a free software for Speaker Verification. In *Odyssey-04 The ISCA Speaker and Language Recognition Workshop*, pages 145–148, Toledo, Spain, May 31- June 3.

Brand, M. (1996). Coupled Hidden Markov Models for modeling interacting Process. Technical report, MIT Lab For Perceptual Computing.

Buckaert, R. R. (1994). A Stratified Simulation Schema for Inference in Bayesian Belief Networks. *Uncertainty in Artificial Intelligence*, pages 110–117.

Buntine, W. (1996). A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, **8**(2), 195–210.

Calado, P., da Silva, A. S., Laender, A. H. F., Ribeiro-Nieto, B. A., and Vieira, R. C. (2004). A Bayesian network approach to searching Web databases through keyword-based queries. *Information Processing and Management*, **40**, 773–790.

Campdell, J. P., Nakasone, H., cieri, C., Miller, D., Walker, K., Martin, A. F., and Przybocki, M. A. (2004). The MMSR Bilingual and Crosschannel Corpora for Speaker Recognition Research and Evaluation. In *Odyssey-04 The ISCA Speaker and Language Recognition Workshop*, pages 29–32, Toledo, Spain, May 31 - June 3.

Carey, M. J., Parris, E. S., Lloyd-Thomas, H., and Bennett, S. (1996). Robust Prosodic Features for Speaker Identification. *International Conference on Spoken Language Processing*, **3**, 1800–1803.

Castillo, E., Gutiérrez, J. M., and Hadi, A. S. (1997). *Expert Systems and Probabilistic Network Models*. Springer Verlag, New York.

Chickering, D. and Heckerman, D. (1997). Efficient Approximations for the Marginal Likelihood of Bayesian Networks with Hidden Variables. *Machine Learning*, **29**, 181–212.

Cooper, G. F. (1989). The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks. *Artificial Intelligence*, **42**, 393–405.

Cooper, G. F. and Herskovits, E. (1992). A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning*, **9**, 309–347.

Dagum, P. and Laby, M. (93). Approximating Inference in Bayesian Networks is NP-Hard. *Artificial Intelligence*, **60**, 141–153.

Daoudi, K., Fohr, D., and Antoine, C. (2000). A New Approach for Multi-Band Speech Recognition based on Probabilistic Graphical Model. In *International Conference on Spoken Language Processing (ICSLP)*.

Daoudi, K., Fohr, D., and Antoine, C. (2003). Dynamic Bayesian Networks for multi-band Automatic Speech Recognition. *Computer Speech and Language*, **17**, 263–285.

DeGroot, M. (1970). *Optimal Statistical Decision*. McGrawHill, New York.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1997). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, **34**, 1–38.

Dielmann, A. and Renals, S. (2004). Dynamic Bayesian Networks for Meeting. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume V, pages 629–632.

Díez, F. J. (1996). Local Conditioning in Bayesian Networks. *Artificial Intelligence*, **87**, 1–20.

Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. Wiley-Interscience.

Faúndez-Zanuy, M. and Rodríguez-Porcheron, D. (1998). Speaker Recognition Using Residual Signal of Linear and Nonlinear Prediction Models. *ICSLP*, **2**, 121–124.

Fernandez, R. and Picard, R. W. (2003). Modeling drivers' speech under stress. *Speech Communication*, **40**, 145–159.

Fisher, D. and Lenz, H. J. (1996). *Learning from Data: Artificial Intelligence and Statistics V . (Lecture Notes in Statistics) vol 112*. Springer Verlag, New York.

Fung, R. and Chang, K. (1990). *Weighing and Integrating Evidence for Stochastic Simulation in Bayesian Networks*, volume 5. In Henrion, M., Shachter, R. D. Kanal, L.N., and Lemmer, J.F. editors, Uncertainty in Artificial Intelligence, North Holland, Amsterdam.

Gauvain, J. L. and Lee, C. H. (1994). Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. *IEEE Trans. Speech Audio Processing*, **2**, 291–298.

Ghahramani, Z. and Jordan, M. I. (1997). Factorial Hidden Markov Models. *Machine Learning*, **29**, 245–273.

Gibbs, W. (1902). *Elementary Principles of Statistical Mechanics*. Yale University Press, NewHaven, Connecticut.

Gilles, D. (2002). Causality, Propensity, and Bayesian Networks. *Synthesis, International Journal for Epistemology, Methodology and Philosophy of Sciences*, **132**, 63–88.

Goldberg, R. and Riek, L. (2000). *A Practical Handbook of Speech Coders*. CRC Press, New York.

Gowdy, J. N., Subramanya, A., Bartels, C., and Bilmes, J. (2004). DBN based Multi-Stream Models for Audio-Visual Speech Recognition. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume I, pages 993–996.

Heckerman, D. (1998). *A tutorial on Learning with Bayesian Network Structures. Learning in Graphical Models*. MIT Press, Cambridge.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, **20**(3), 197–243.

Henrion, M. (1988). Propagating Uncertainty by Logic Sampling in Bayes' Networks. *Uncertainty in Artificial Intelligence*, pages 317–324.

Henrion, M. (1991). Searched-Based Methods to Bound Diagnostic Probabilities in Very Large Belief Nets. In *In Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 142–150, San Mateo, California.

Hershey, J., Attias, H., Jojic, N., and Kristjansson, T. (2004). Audio-visual Graphical Models for Speech Processing. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume V, pages 649–652.

Huang, C. and Darwiche, A. (1994). Inference in Belief Networks: Procesural Guide. *International Journal of Approximate Reasoning*, **11**, 1–158.

Jensen, F. and Andersen, S. K. (1990). Approximation in Bayesian Belief Universes for Knowledge Based Systems. *Proceedings of the 6th Workshop on Uncertainty in Artificial Intelligence, Cambridge, MA*.

Jensen, F. V. (1996). *Bayesian Networks and Decision Graphs*. Springer, New York.

Kashino, K. and Murase, H. (1999). A sound identification system for ensemble music based on template adaptation and music stream extraction. *Speech Communication*, **27**, 337–349.

Kenny, P., Lennig, M., and Mermelstein, P. (1990). A Linear Prediction HMM for Vector-Valued Observations with Applications to Speech Recognition. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 38, pages 220–225.

Kim, J. H. and Pearl, J. (1983). A Computational Model for Combined Causal and Diagnostic Reasoning in Inference System. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 190–193.

Kjaerulff, U. (1993). Approximation of Bayesian Networks Through Edge Removals. Technical report, Dept. Mathematics and Computer Science, Aalborg University.

Kschischang, F. R., Frey, J. B., and Loeliger, H. A. (2001). Factor Graphs and The Sum-Product Algorithm. *IEEE Transactions on Information Theory*, **47**(2), 498–519.

Lam, W. and Bacchus, F. (1994). Learning Bayesian Belief Networks. An Approach Based on the MDL Principle. *Computational Intelligence*, **10**(4).

Lauritzen, S. L. (1995). The EM Algorithm for Graphical Association Models with Missing Data. *Computational Statistics and Data Analysis*, **19**, 191–201.

Lauritzen, S. L. (1996). *Graphical Models*. Oxford Science Publications, Oxford.

Lauritzen, S. L. and Spiegelhalter, D. J. (1988). Local Computation with Probabilities on Graphical Structures and Their Application to Expert systems. *Journal of the Statistical Royal Society*.

Leggetter, C. J. and Woodland, P. C. (1995). Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models. *Computer Speech and Language*, **9**, 171–185.

Li, K. P. and Porter, J. E. (1988). Normalization and selection of speech segments for speaker recognition scoring. *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP'88)*, **1**, 595–598.

Linde, Y., Buzo, A., and Gray, R. M. (1980). An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications*, **28**, 84–95.

Markov, K. and Nakamura, S. (2003). Hybrid HMM/BN LVCSR System Integrating Multiple Acoustic Features. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume I, pages 840–843.

Matsui, T. and Furui, S. (1994). Speaker Adaptation of Tied-mixture-based Phoneme Models for Text-prompted Speaker Recognition. *Proceedings ICASSP*, pages 1125–1128.

McEliece, R. J., MacKey, D. J. C., and Cheng, J. F. (1998). Turbo Decoding as an Instance of Pearl's 'Belief Propagation' Algorithm. *IEEE in Selected Areas Communication*.

Mistretta, W. J. and Farrell, K. R. (1998). Model Adaptation Methods for Speaker Verification. *Proceedings ICASSP*.

Mokbel, C. (2001). Online Adaptation of HMMs to Real Life Conditions: A Unified Framework. *IEEE Trans. on Speech and Audio Processing*.

Moore, B. C. J. (1982). *An Introduction To The Psychology of Hearing*. Academic Press.

Murphy, K. P. (2001). The Bayes Net Toolbox for Matlab. *Computing Science and Statistics: Proceedings of the Interface*, **33**.

Murphy, K. P. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. thesis, University of California, Berkeley.

Naik, J. M. and Doddington, G. R. (1986). High Performance Speaker Verification Using Principal Spectral Components. *Proceedings ICASSP*, pages 881–884.

Nefian, A., Lian, L., Pi, X., Xiaoxiang, L., Mao, C., and Murphy, K. P. (2002). A Coupled HMM for Audio-Visual Speech Recognition. *Int. Conference on Acoustics, Speech and Signal Proc.*

NIST's 2003 Speaker Recognition Evaluation (2003). http://www.nist.gov/speech/tests/spk/2003/index.htm.

NIST's 2004 Speaker Recognition Evaluation (2004). http://www.nist.gov/speech/tests/spk/2004/index.htm.

NIST's Speaker Recognition Evaluation (2004). http://www.nist.gov/speech/tests/spk/.

Pearl, J. (1987a). Distributed Revision of Compatible Beliefs. *Artificial Intelligence*, **33**, 173–215.

Pearl, J. (1987b). Evidencial Reasoning Using Stochastic Simulation of Causal Models. *Artificial Intelligence*, **42**.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Diego.

Pelecanos, J. and Sridharan, S. (2001). Feature Warping for Robust Speaker Verification. In *2001: A Speaker Odyssey. The Speaker Recognition Workshop*, Crete, Greece, June 18-22.

Poole, D. (1993). Average-Case Analysis of a Search Algorithm for Estimating Prior and Posterior Probabilities in Bayesian Networks with Extreme Probabilities. In *In Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 606–612, San Mateo, California.

Prasanna, S. R. M., Zachariah, J. M., and Yegnanayayana, B. (2004). Neural Networks Models for Combining Evidence from Spectral and Suprasegmental Features for Text-Dependent Speaker Verification. In *Proceedings of International Conference on Intelligent Sensing and Information Processing ICISIP, IEEE*, pages 359–363.

Reynolds, D. A. (1996). The Effect of Handset Variability on Speaker Recognition Performance: Experiments on the Switchboard corpus. In *ICASSP'86*, volume 1, pages 113–116, Atlanta, Ga, USA.

Reynolds, D. A. (1997). Comparison of Background Normalization Methods for Text Independent Speaker Verification. *Eurospeech*.

Reynolds, D. A. (2000). Score Normalization for Text-Independent Speaker Verification System. *Digital Signal Processing*, **1**.

Robinson, R. W. (1977). *Counting unlabeled acyclic Diagraphs*, volume 622 of *Lecture Notes in Mathematics*. Berlin, Springer C.H.C. LITTLE, Ed.

Robles, V., Larrañaga, P., Peña, J. M., Menasalvas, E., Pérez, M. S., Herves, V., and Wasilewska, A. (2004). Bayesian network multi-classifiers for protein secondary structure prediction. *Artificial Intelligence in Medicine*, **31**, 117–136.

Sánchez-Soto, E., Blouet, R., Chollet, G., and Sigelle, M. (2003). Speaker Verification with Bayesian Networks. In *Workshop on Multimodal User Authentication*, pages 61–65, Santa Barbara, California, December 11-12.

Sánchez-Soto, E., Sigelle, M., and Chollet, G. (2004a). Graphical Models for Text Independent Speaker Verification. In *International School on Nonlinear Speech Processing*, pages 27–31, Vietri Sul Mare, Italy, September 13-18.

Sánchez-Soto, E., Blouet, R., Sigelle, M., and Chollet, G. (2004b). Model Adaptation for Speaker Verification using Conditional Probability Tables. In *Workshop on Biometrics on the Internet (COST 275)*, pages 27–31, Vigo, Spain, March 25-26.

Sang, L., Wu, Z., Yang, Y., and Zhang, W. (2003). Automatic Speaker Recognition Using Dynamic Bayesian Network. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume I, pages 188–191.

Shachter, R. and Kenley, C. (1989). Gaussian Influence Diagrams. *Management Science*.

Shachter, R. and Peot, M. (1990). *Simulation Approaches to General Probabilistic Inference on Belief Networks*, volume 5. In Henrion, M., Shachter, R. D. Kanal, L.N., and Lemmer, J.F. editors, Uncertainty in Artificial Intelligence, North Holland, Amsterdam.

Sigelle, M. (2003). Tree-Structured Probabilistic Graphical Models. Unpublished Technical Report, École Nationale Supérieure des Télécommunications, Paris.

Spirtes, P., Glymour, C., and Scheiner, R. (1993). *Causation, Prediction and Search*. MIT Press, Boston, 2nd edition.

Srinivas, S. and Nayak, P. (1996). Efficient Enumeration of Instantiations in Bayesian Networks. In *In Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 500–508, San Mateo, California.

Stephenson, T. A., Bourlard, H., Bengio, S., and Morris, A. C. (2000). Automatic speech recognition using dynamic Bayesian networks with both acoustic and articulatory variables. In *6th International Conference on Spoken Language Processing: ICSLP 2000 (Interspeech 2000)*, pages II:951–954, Beijing.

Stephenson, T. A., Doss, M. M., and Bourlard, H. (2001). Modeling Information in Bayesian Networks Based ASR. *European Conference on Speech Communication an Technology (Eurospeech)*.

Stetter, M., Deco, G., and Dejori, M. (2003). LArge-Scale Computational Modeling of Genetic Regulatory Networks. *Artificial Intelligence*, **20**, 75–93.

Switchboard Corpora LDC (2004). http://www.ldc.upenn.edu/.

The ELISA Consortium (2004). http://www.lia.univ-avignon.fr/heberges/alize/elisa/index.html.

Thévenaz, P. (1993). *Résidu de Prédiction Linéaire et Reconnaissance de Locuteurs Indépendante du Texte*. Ph.D. thesis, Université de Neuchâtel Institut de Microtechnique.

Verma, T. and Pearl, J. (1991). *Equivalence and Synthesis of Causal Models*. Uncertainty in Artificial Intelligence, Amsterdam.

Wan, V. and Campbell, W. M. (2000). Support Vector Machines for Speaker Verification and Identification. In *signal Processing Neural Networks for Signal Society Workshop Processing, IEEE*, volume 2, pages 775–784, Crete, Greece, December 11-13.

Weber, K., Bengio, S., and Bourlard, H. (2002). Increasing Speech Recognition Robustness with HMM2. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume I, pages 929–932.

Wellekens, C. J. (1987). Explicit Time Correlation in Hidden Markov Models for Speech Recognition. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 12, pages 384–386.

Wold, H. D. A. (1960). A Generalization of Causal Chain Models. *Econometrica*, **28**, 443–63.

Wright, S. (1921). Correlation and Causation. *Journal of Agricultural Research*, **20**, 557–85.

Zweig, G. (1998). *Speech Recognition with Dynamic Bayesian Networks*. Ph.D. thesis, U. C. Berkeley.