
Ircam - Equipe MuTant (Inria)

# Adaptation au locuteur pour le suivi de voix chantée

*Auteur :*
Stephane Rivaud

*Encadrants :*
Nicolas Obin
Philippe Cuvillier

11 août 2015

## Résumé

Le suivi de partition est un outil permettant à une machine d'écouter et de se synchroniser avec une performance musicale dont on lui a fournit la partition au préalable. Cette tache extrêmement délicate est cruciale pour qu'un ordinateur puisse intéragir avec un orchestre et soulève bien des problèmes. Parmis ces problèmes on trouve l'étude des instruments que l'on doit suivre afin d'apporter à l'ordinateur un maximum de connaissance sur le son qu'il doit écouter. Dans le cas de la voix chantée, nous devons prendre en compte l'aspect symphonique du chant à qui nous attribuons une partition classique, ainsi que les paroles de ce chant. Comparées aux instruments de factures, la voix est dotée d'une énorme quantité de paramètres, continus pour la plupart, que le chanteur (professionnel ...) manipule avec rapidité et précision. Cette variabilité sonore rend le suivi de voix chantée extrêmement délicat. Dans ce stage, nous allons chercher à traiter l'information phonétique que l'on verra comme une interprétation du texte de la chanson.

# Table des matières

# Chapitre 1

# Introduction

## 1.1 Le suivi de partition

Le suivi de partition consiste, étant donné une partition, à inférer en temps-réel lors d'une performance musicale la position courante dans la partition. Bien que les informations contenues dans la partition soient véhiculéew par le signal sonore produit par les musicien, il s'agit d'informations ayant un haut niveau d'abstraction. De plus, le signal sonore est continu, tandis qu'une partition est un ensemble discret. Afin ce résoudre le problème du suivi de partition, nous allons devoir extraire des informations symboliques du signal audio entrant, puis les combiner afin de décider, à chaque pas de temps, quelle note est en train d'être jouée.

L'architecture du suivi de partition qui est implémenté dans antescofo a été élaboré dans [1]. Le signal acoustique de la performance musicale est modélisé comme une chaine de semi-Markov, où la séquence cachée est la partition de la pièce jouée. Les chaines de semi-Markov sont une extension des chaines de Markov où la distribution de la durée d'occupation d'un état est modélisée par une loi, qui devient un paramètre du modèle. Dans les modèles de Markov cachés, cette loi est implicitement une loi géométrique. Cette extension est particulièrement bien adaptée au suivi de partition car nous avons un apriori très fort sur les durées d'occupations des états qui nous est données par la partition.

Comme dans les modèles de Markov, les modèles de semi-Markov nécessitent également une probabilité d'observation d'un évènement conditionnellement à un état caché. Cette probabilité d'observation est construite selon l'instrument que l'on va suivre : dans le cas d'un instrument harmonique, cette probabilité peut être construite à partir d'une information de hauteur fondamentale. La modélisation d'une caractéristique sonore d'un instrument permet au logiciel d'être beaucoup plus précis dans son processus d'inférence. Prenons l'exemple d'un instrument percussif tel qu'un piano : la présence d'un saut d'énergie au début de chaque note permet de détecter les onsets de façon robuste, et donne donc une précision temporelle accrue à notre algorithme. Dans le cas des signaux de chants, nous avons beaucoup moins d'apriori quant à la structure temps-fréquence que l'on pourrait obtenir étant donnée une partition classique.

Mon stage a donc pour but de proposer une amélioration du système de suivi de partition Antescofo dans le cadre du suivi de voix chantée.

## 1.2 La voix chantée

Afin d'enrichir notre modèle de voix chantée, nous allons inclure le texte de la chanson à notre partition. Les travaux de Rong Gong (stagiaire de l'équipe MuTant 2013-2014) montrent qu'il est possible d'avoir un modèle spectral d'une voyelle chantée beaucoup plus élaboré si cette voyelle est connue à l'avance [6]. Bien que l'on puisse définir un modèle spectral joint combinant information de hauteur et information phonétique, il apparait que la fusion tardive de ces deux informations dans le processus d'inférence donne de meilleur résultats. A l'instar de ces travaux, nous avons cherché à modéliser l'information phonétique de façon indépendante afin d'envisager une fusion tardive des informations de hauteur et phonétiques.

Dans l'optique de modéliser l'information phonétique véhiculée par la voix chantée et d'effectuer un alignement avec les paroles, nous avons exploré la littérature d'alignement audio to lyrics. Bien qu'elle soit assez parcimonieuse, on remarque que celle-ci traite le problème dans deux principales situations : lorsque le signal de parole est déjà séparé [9], et lorsque l'accompagnement est présent dans le signal audio [11], [12], [7] et [10].
Lorsque l'accompagnement est présent, beaucoup d'efforts sont investis dans la séparation de source pour extraire ou estimer le signal de chant. Les travaux de [9] sont ceux qui se rapprochent le plus de notre objectif, en proposant un alignement phonème par phonème et en proposant une architecture optimisée pour le temps-réel. Ces travaux sont axés sur l'élaboration d'un espace d'états pour HMMs optimisé pour le suivi de voix chantée, et l'élaboration d'un algoritme Viterbi optimisé pour éviter le backtracking afin de minimiser le délai entre la captation du signal audio et l'inférence de la position courante dans la partition.

Antescofo étant confronté à des signaux de paroles théoriquement débarassés de l'accompagnement, nous nous sommes attaqué au problème de l'alignement Lyrics to text via la littérature de reconnaissance de parole. Cette approche est quelque peu entammée par les travaux cités précedemment dans le cadre de l'alignement des signaux de chant avec accompagnement. Notre cadre soulève des contraintes différentes, et les améliorations que nous proposons sont de tout autre nature.

## 1.3 Démarche

### 1.3.1 Littérature de Reconnaissance de parole

La reconnaissance de parole est un domaine qui suscite un très grand intérêt de la part de la communauté de traitement du signal audio. La première chose que l'on remarque dans cette littérature est que les modèles de voix parlées sont les mêmes que les modèles de suivi de partition : le signal audio est supposé être une chaine de markov cachés. Plusieurs modèles d'états cachés existent, les plus utilisés étant les modèles hiérarchiques et les modèles de phonèmes en contexte qui permettent de prendre en compte les phénomènes de coarticulation. Mais la reconnaissance de parole se heurte à plusieurs problèmes de taille.

Lors d'un échange verbale on ne connait pas à priori la vitesse à laquelle un locuteur parle, si tant est que l'on puisse définir une telle quantité. Ce problèmes lié à la distribution temporelle d'occupation des états cachés est l'un des plus difficile à surmonter.

L'abscence d'information à priori sur les mots qui sont prononcés lors d'un dialogue est un autre problème. Une solution est d'avoir un dictionnaire de mots possibles, et de proposer une séquence cachée possible avec une mesure de confiance, puis de rechercher le mot correspondant le plus probable dans le dictionnaire. La connaissance du sujet, du type de tournure de phrase employée par le locuteur permet d'améliorer les performances. Mais ce problème n'a guère à voir avec la modélisation acoustique de notre signal.
Ces deux problèmes étant déjà résolus grâce à l'architecture d'Antescofo, nous ne nous concentrerons donc pas là-dessus.

Le troisième problème majeur rencontré dans la reconnaissance de parole est la variabilité acoustique rencontrée dans le signal. Elle est majoritairement dûe à deux choses : les différences entre les locuteurs et les différences de conditions acoustiques d'enregistrement du signal audio.
Les variabilités de conditions acoustiques sont surtout dûes aux bruits environnants ainsi qu'à la diversité des capteurs. En revanche les variabilités dues aux locuteurs sont de deux types : les variabilités intra-locuteurs et les variabilités inter-locuteurs.
Les variabilités intra-locuteur reflètent la diversité et la richesse des sons que l'on peut obtenir avec un appareil de production vocale. C'est ainsi que l'on peut arriver à modeler un langage parlé et même une voix chantée. Cette variabilité est le coeur de la modélisation acoustique de la voix parlée.
Les variabilités inter-locuteur quant à elles sont un obstacle à la modélisation. S'il est déjà difficile d'obtenir un modèle spécifique à un locuteur, il l'est encore plus d'avoir un modèle qui puisse convenir à tous les locuteurs. Les modèles locuteurs-indépendants ont une performance moindre en reconnaissance de parole, et nécessitent de grandes bases de données d'apprentissages. Mais ces modèles étant indépendant du locuteur, les sources de la base de données d'apprentissage peuvent (et doivent) être variées.
Les modèles locuteur-dépendants sont quant à eux très performants mais requièrent de grandes bases de données d'apprentissages homogènes, ce qui est difficile à obtenir à partir d'un locuteur unique. Cette problématique de base de données d'apprentissage est cruciale sur les systèmes de reconnaissance vocales "en-ligne" tels que présents sur les smartphones, où les données d'apprentissage arrivent au fil de l'eau. Ceci a poussé à l'élaboration de modèle locuteur-adaptatif.

### 1.3.2 Besoin d'adaptation rapide et robuste

Les modèles locuteur-adaptatifs sont des modèles qui ont pour but d'atteindre les performances d'un système locuteur dépendant avec très peu de données d'apprentissage. Pour ce faire, on entraine au préalable un modèle que l'on sait être robuste, puis on calcul une transformation des paramètres de ce modèle à partir de données provenant du locuteur ciblé. Ce type de modèle devient rapidement nécessaire dans les modules de reconnaissance de parole des interfaces homme-machine où les locuteurs ne sont jamais les mêmes (ex : *hotline téléphoniques, reconnaissance vocale sur les smartphone,...*). Plusieurs méthodes ont alors été étudiées, on peut en trouvé un résumé dans [15].

La première méthode pourrait être de réapprendre les paramètres du modèle d'observation grâce à l'algorithme EM classique d'apprentissage des paramètres d'une HMM (Baum-Welch). La difficulté de cette approche est que la fonction de vraissemblance des données comporte d'autant plus de maximum locaux qu'il y a de paramètres. L'estimation de tous ces paramètres avec peu de données d'adaptation pourrait donner de moins bon

résultats que l'utilisation d'un modèle non adapté.

La seconde méthode d'adaptation étudiée dans la littérature fut celle du Maximum à Posteriori (MAP). Elle permet d'améliorer l'aspect non-supervisé de l'estimation EM classique : l'apriori est donné par le modèle appris au préalable. Toutefois, seuls les paramètres des classes observées sont mis à jour, il faut donc observer toutes les classes avant d'avoir mis à jour tous les paramètres. Cela nécessite toujours beaucoup de données d'adaptation, surtout dans le cas où certains évènement sont rares. De plus, lors d'une réestimation MAP, on oublie les paramètres appris hors-ligne, et on perd l'information contenu dans le modèle de référence.

Afin de contourner ce problème, une méthode basée sur la correlation entre les paramètres fut développée : à l'aide d'un grand corpus de parole, on examine les corrélations entre les paramètres du modèles losque l'on fait varier les locuteurs. Si deux paramètres sont fortement corrélés, alors on leur applique la même transformation lors de l'adaptation. Ceci permet d'adapter des paramètres relatifs à un évènement encore non observé, en contraignant notre ensemble de paramètres à respecter certaines relations [2], [3].

Une deuxième façon de contraindre l'espace des paramètres est de le structurer dans un Regression Class Tree, et d'appliquer des transformations selon le critère MAP de façon hiérarchique. Ainsi à la racine de l'arbre, on estime une transformation globale, puis on se sert de cette transformation global comme à apriori pour estimer les transformations dans les sous-arbres. Cette technique permet une adaptation plus robuste avec moins de données d'adaptation [14].

La dernière méthode d'adaptation proposée dans la littérature est celle du Maximum Likelihood Linear Regression. Nous contraignons l'ensemble des paramètres du modèle adapté en supposant qu'il est l'image de l'ensemble des paramètres du modèle de référence par une transformation affine. Cette méthode eu un rapide succès grâce à sa robustesse et du peu de données d'adaptation nécessaires à l'estimation de ces transformations. De plus, on peut contraindre l'ensemble des transformations possibles afin de réduire le nombre de paramètre à estimer. C'est cette méthode que nous avons explorée durant ce stage.

# Chapitre 2

# Modélisation de la loi d'observation

## 2.1 Les phonèmes

La phonème est la plus petite unité linguistique véhiculant de l'information dans un signal de parole. Afin de situer la hiérarchie de la construction d'une phrase :

— une phrase est consituée de mots. ( ex : "je - vais - à - là - marina" )
— un mot est composé de syllabes. ( ex : "ma - ri - na" )
— une syllabe est composée de phonèmes. ( ex : "ma" = 'm' - 'a' )

Il y a environ 35 phonèmes dans la langue française (la prise en compte ou non de certaines semi-voyelles fait varier ce chiffre). L'alphabet phonétique est à un mot parlé ce que sont les symboles du solfège à une composition musical. Nous allons donc étudier les phonèmes, qui sont ici l'interprétation sonore des symboles phonétiques.

## 2.2 Description de la base de données

Afin de tester les performances des algorithmes présents dans la littérature de reconnaissance de parole dans le cadre du suivi de voix chantée, nous avons utilisé une base de données fournie par l'équipe Analyse-Synthèse de l'Ircam. Il s'agit d'une base de données élaborée suivant le protocole suivant :

1. Choisir une liste de mots telle que l'ensemble des mots regroupent tous les phonèmes de la langue française.
2. Enregistrer 2 locuteurs, un homme et une femme, chantant chaque mot de la liste à une hauteur constante.

Lors de l'enregistrement de cette base de données, le locuteur femme a enregistré chaque mot dans deux octaves différents, l'un dans un style "musique populaire" et l'autre dans un style "chant lyrique". Nous considérerons donc que nous avons 3 locuteurs pour cette base de données. Les enregistrements sont au format wav, échantillonnés à 48000hz, et codé sur 24-bits. Cela constitue donc une base de chant mono-hauteur nous permettant d'étudier la voix chantée dans des conditions réelles mais quasi-idéales, ainsi que les variations sonores entre un homme et une femme interprêtant la même chanson. La durée totale d'enregistrement par locuteur varie entre 45 et 60 minutes : ceci constitue une large base de données.

Le descripteur majoritairement utilisé dans la reconnaissance de parole étant les MFCCs, j'ai calculé des MFCCs à partir de cette base de données de chant mono-hauteur avec les spécifications suivantes :

1. Calcul du spectrogramme de chaque morceau avec les paramètres suivants
   — Un fenêtrage de 1024 échantillons pondérés par une fenêtre de Hanning.
   — Un taux de recouvrement de 75%.
   — Le carré du module de la FFT de chaque trame est calculé sur 1024 points.
2. Passage de l'échelle linéaire à l'échelle de Mel avec 40 bandes de Mel.
3. Application d'une compression logarithmique.
4. Calcul de la transformée en cosinus discrète sur 40 points.

La voix est modélisée à l'aide d'un modèle source filtre. La source est l'excitation glottique et le filtre ou résonateur est le conduit vocal, le signal de parole est donc une convolution de l'excitation glottique par la réponse impulsionnelle du résonateur. Les MFCCs, via la compression logarithmique, ont la bonne propriété d'effectuer une déconvolution du signal : les premiers coefficients MFCCs représentent la contribution du résonateur, tandis que les derniers coefficients représentent la contribution de l'excitation glottique. Nous faisons l'hypothèse que la contribution du résonateur contient l'information phonétique, ainsi nous ne conservons que les 13 premiers coefficients.

En plus de ces enregistrements mono-hauteur à partir desquels nous avons calculé les MFCCs, nous disposons d'une segmentation de ces fichiers.wav en phonèmes successifs. Ceux-ci sont présents sous la forme d'annotations au format *sdif* dans la base de données. Celà nous donne en quelque sorte la partition phonétique associée à l'enregistrement du mot en question.

## 2.3 Variabilité intra-locuteur

Afin de visualiser les variabilités intra-locuteurs, j'ai effectué quelques histogrammes,pour avoir une idée des différentes densités que l'on doit approximer. On se rend compte que pour une voyelle donnée, la densité ressemble à une gaussienne sur chaque dimension. On peut donc raisonnablement supposer que nous pouvons approximer la distribution d'un descripteur conditionnellement à un phonème par une gaussienne.

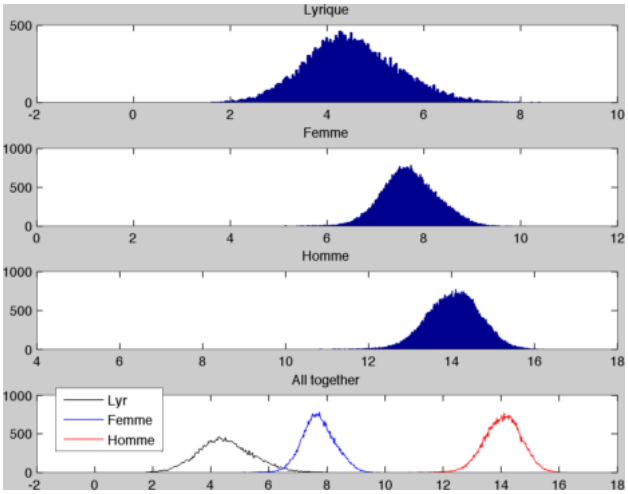



FIGURE 2.1 – Histogrammes fait à partir d'observation des trois corpus du phonème `'a'`, sur le deuxième coefficient MFCC.

Ce constat n'est déjà plus très valable sur les distribution conditionnelle des consonnes. Par exemple, le `'d'` a une distribution plus chaotique, qui nécessiterait probablement plusieurs modes pour être représentée correctement. Cela peut être dû à des phénomène de co-articulation, où à des erreurs dans les annotations étant donné la brièveté temporelle de cet évènements : ces autres modes proviendraient dans ce cas de la distribution d'un autre phonème.

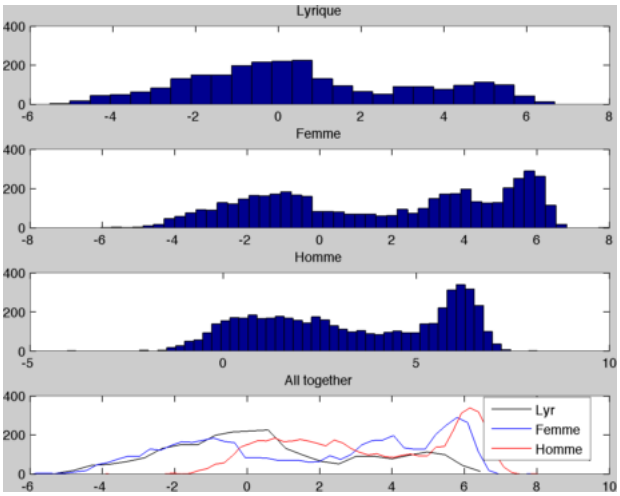



FIGURE 2.2 – Histogrammes du troisième coefficient MFCC réalisés à partir d'observation du phonème `'d'`

Nous allons donc nous concentrer sur l'estimation des modèles de reconnaissance de voyelle, non seulement parce que les modèles semblent plus facile à approximer, mais aussi parce que les voyelles représentent plus de 75% de l'activité vocalique.

## 2.4 hypothèses de modélisation

Etant donné la variabilité que l'on observe dans la voix humaine, et l'abscence de modèle physique permettant de décrire et d'étudier de manière fiable les signaux de paroles, nous allons nous tourner vers une approche probabiliste.

Supposons que le descripteur MFCC est une variable aléatoire, nous allons chercher à calculer sa distribution de probabilité. L'hypothèse généralement faite en reconnaissance de parole est que cette loi, conditionnellement à un locuteur, est une mixture de gaussienne ayant autant de mode qu'il existe de phonème.

Comme la base de données contient des annotations *sdif* représentant la transcription du signal en phonème, nous disposons d'une base de données étiquetée. Nous pouvons donc apprendre les paramètres de ce GMM par apprentissage supervisé.

### 2.4.1 Protocole d'apprentissage

Notons $O$ la variable aléatoire représentant les MFCCs d'un locuteur donné. Il s'agit d'une variable aléatoire à valeurs dans $\mathbb{R}^{13}$ dont la distribution est une mixture de 35 gaussiennes. Nous avons donc 35 moyennes et 35 matrices de covariance à estimer. Comme nous disposons d'une base de données étiquetée, nous pouvons estimer chaque moyenne et chaque covariance avec l'estimateur du maximum de vraissemblance.

Soit $\mathcal{M} = (\mu_s, \Sigma_s)_{1 \leq s \leq 35}$ l'ensemble des paramètres de la mixture, on pose pour tout $s \in \mathbb{N}$ tel que $1 \leq s \leq 35$ :

$$\mu_s = \frac{\sum_{S(o_t)=s} o_t}{\#\{S(o_t) = s\}}$$

$$\Sigma_s = \frac{\sum_{S(o)=s} (o_t - \mu_s)(o_t - \mu_s)^\top}{\#\{S(o) = s\}}$$

où l'ensemble $(o_t)_{t \in [|1,T|]}$ est l'ensemble des observations pour un locuteur donné. Le nombre d'observations utilisées pour l'estimation varie entre 9000 et 70000 pour les voyelles, et est de l'ordre de 5000 observations pour les semi-voyelles et les consonnes qui ont une durée plus brève.

Ainsi, nous avons estimé $\mathcal{M}$ grâce à notre base de données, ce qui nous permet de calculer une approximation de la distribution de $O$ conditionnellement à un locuteur.

### 2.4.2 Variabilité inter-locuteur

Afin d'illustrer les variabilités inter-locuteurs, j'ai visualisé sur la même figure, les modèles de 3 locuteur différents, marginalisés suivant la même dimension pour un phonème donné. En l'occurence il s'agit du 7-ème coefficient MFCC pour le phonème `'i'` dans la figure ci-dessous. On remarque qu'il y a une véritable variation des paramètres des modèles en fonction du locuteur (ceci n'est pas un cas isolé). Le développement de stratégie d'adaptation s'avère donc nécessaire.

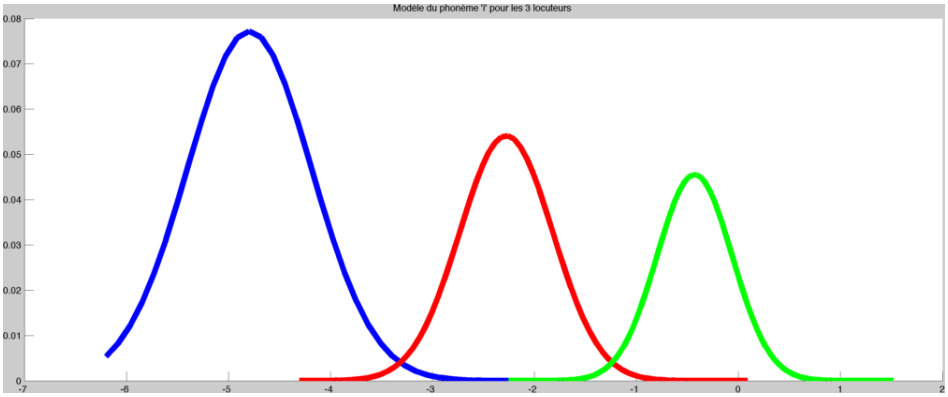



FIGURE 2.3 – Gaussiennes modélisant la distribution du 7-eme coefficient MFCC conditionnellement au phonème `'i'` pour les trois locuteurs du corpus.

# Chapitre 3

# Adaptation par Maximum Likelihood Linear Regression

## 3.1 Etat de l'art

Afin de compenser les variabilités inter-locuteur dans les modèles de reconnaissance de paroles, des stratégies d'adaptation de modèles furent développées. La plus couramment utilisée aujourd'hui est celle du Maximum Likelihood Linear Regression (MLLR). Il existe trois grands types d'adaptation dans la littérature consacrée : l'adaptation des moyennes seules, l'adaptation des moyennes et des covariances séparément, et l'adaptation contrainte des moyennes et des covariances avec la même transformation.

Le premier algorithme d'adaptation présenté dans [8], consiste à estimer une transformation affine des paramètres d'un modèle afin de maximiser la vraissemblance des données d'adaptation. En notant $\mathcal{M} = (\mu_s, \Sigma_s, \alpha_s)_{1 \leq s \leq S}$ les paramètres du modèle initial, les paramètres du modèle adapté sont de la forme $\hat{\mathcal{M}} = (\hat{A}\mu_s + \hat{b}, \Sigma_s, \alpha_s)_{1 \leq s \leq S}$ avec $\hat{A} \in \mathcal{M}_{d,d}(\mathbb{R})$ et $\hat{b} \in \mathbb{R}^d$ où $d$ est la dimension du vecteur d'observation.

Une extension de cet algorihme fut ensuite présentée dans [5] afin de pouvoir adapter les matrices de covariances. Les nouveaux paramètres $\hat{\mathcal{M}} = (\hat{\mu_s}, \hat{\Sigma_s}, \alpha_s)_{1 \leq s \leq S}$ sont maintenant de la forme

$$\hat{\mu_s} = A\mu_s + \hat{b}$$
$$\hat{\Sigma_s} = L_s \hat{H} L_s^\top$$

avec $L_s = B_s^{-1}$ où $B_s$ est la matrice triangulaire inférieure dans la décomposition de cholesky de $\Sigma_s$, i.e. $\Sigma_s = B_s B_s^\top$.

Dans l'adaptation des moyennes et des variances, l'estimation de la transformation $\hat{H}$ était trop couteuse pour être implémenté dans un cadre temps-réel, et une nouvelle forme d'adaptation fut mise au point : le MLLR-contraint, présenté dans [4].

Cette fois-ci, le nouveau jeu de paramètres $\hat{\mathcal{M}} = (\hat{\mu_s}, \hat{\Sigma}_s, \alpha_s)_{1 \leq s \leq S}$ est de la forme

$$\hat{\mu_s} = A\mu_s + \hat{b}$$
$$\hat{\Sigma}_s = \hat{\Sigma}_s = A\Sigma_s A^\top$$

Cette forme d'adaptation est aussi appelée *feature*-MLLR, car on peut l'interpréter comme une adaptation des observations afin de maximiser leur vraissemblance. La transformation des observation équivalente à la transformation des paramètres est

$$\tilde{O}_t = \tilde{A}O_t + \tilde{b}$$

avec

$$\tilde{A} = A^{-1}$$

et

$$\tilde{b} = -A^{-1}b$$

Par la suite, des améliorations de ces algorithmes ont été proposées afin de réduire le temps de calcul et minimiser les erreurs d'arrondies principalement en évitant les inversions de matrices dans les algorithmes, ainsi que pour s'extraire des hypothèses requisent par les premiers algorithmes, notamment le fait que les matrices de covariances sont supposées être diagonales. Des stratégies d'adaptation plus complexes ont aussi été mises au point afin de compenser l'hétérogénéité dans les bases de données d'entrainement, l'incertitude quand à la transcription des données d'adaptation, ainsi que des stratégies d'adaptation hiérarchique étant donné le grand nombre de paramètres dans les modèles de reconnaissance vocale à grand vocabulaire.
Dans le cadre du suivi de voix chantée, nous n'avons pas été confronté à ces problématiques : nos données d'apprentissage pour un locuteur donné sont homogènes, nous connaissons la transcription de ces données d'apprentissage via la partition, et nous connaissons donc à l'avance notre espace d'état. Nous avons donc expérimenté les premiers algorithmes d'adaptation afin de les confronter au contexte du suivi de voix chantée.

**Remarques théoriques :** Une des forces de l'adaptation par MLLR est de pouvoir adapter un grand nombre de paramètres avec peu de données d'adaptation. Ceci se fait en contraignant une transformation à être la même pour plusieurs classes. Sans cette contrainte, estimer une transformation par classe ne serait pas plus efficace que de simplement réestimer les paramètres.
Il est donc important de connaitre les limites théoriques de cette approche. Plaçons nous dans le cas où toutes les matrices de covariances sont identiques : estimer une transformation en maximisant la vraissemblance des données d'adaptation est équivalent à estimer une transformation en minimisant la somme des erreurs quadratiques avec la distance associée au produit scalaire induit par ces matrices de covariance. Or nous savons que les problèmes de régression linéaire par méthode des moindre carrés admettent une solution exacte si il y a au plus $d + 1$ contraintes, où $d$ est la dimension du vecteur d'observation.
En résumé, adapter plus de $d + 1$ paramètres avec la même transformation impose bel et bien une contrainte sur notre espace de paramètres. Cette contrainte est que si on prend un système de coordonnées barycentriques dans l'ensemble des paramètre du modèle de référence, et que l'on considère le système correspondant dans l'ensemble des paramètres du

modèle cible, alors les coordonnées des paramètres de chaque classe doivent être identiques dans le système de référence et dans le système cible. Cela suppose donc que l'information d'un locuteur donné est contenue dans un espace affine de dimension $d$.

## 3.2 Présentation des algorithmes

### 3.2.1 Moyennes seules

Soit $O = (O_1, \cdots, O_T)$ une suite de variables aléatoires i.i.d. et soit une mixture de gaussienne de paramètres $\mathcal{M} = (\mu_s, \Sigma_s, \alpha_s)_{1 \leq s \leq S}$. On souhaite calculer $\hat{W} \in \mathcal{M}_{d,d+1}(\mathbb{R})$ telle que la mixture de gaussienne de paramètre $\bar{\mathcal{M}} = (W\xi_s, \Sigma_s, \alpha_s)_{1 \leq s \leq S}$ avec $\xi = (\mu_s^\top, 1)^\top$ maximise la vraissemblance de $O$.

La log-vraissemblance des observations s'écrit comme suit :

$$\begin{aligned}
\mathcal{L}(O|\mathcal{M}) &= \log(\mathbb{P}(O|\mathcal{M})) \\
&= \log(\prod_{t=1}^{T} \mathbb{P}(O_t|\mathcal{M})) \\
&= \sum_{t=1}^{T} \log(\mathbb{P}(O_t|\mathcal{M})) \\
&= \sum_{t=1}^{T} \log(\sum_{\theta_t \in \mathcal{S}} \mathbb{P}(O_t|\mathcal{M}, \theta_t)\mathbb{P}(\theta_t|\mathcal{M})) \quad \textit{avec } \mathcal{S} \textit{ l'ensemble des classes} \\
&\geq \sum_{t=1}^{T} \sum_{\theta_t \in \mathcal{S}} \mathbb{P}(\theta_t|\mathcal{M}) \log(\mathbb{P}(O_t|\mathcal{M}, \theta_t)) \\
&\geq \sum_{t=1}^{T} \sum_{\theta_t \in \mathcal{S}} \mathbb{P}(\theta_t|\mathcal{M}) \log(\mathcal{N}(O_t, \mu_s, \Sigma_s))
\end{aligned}$$

avec

$$\mathcal{N}(O_t, \mu_s, \Sigma_s) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_s|}} \exp(-\frac{1}{2}(O_t - \mu_s)^\top \Sigma_s^{-1}(O_t - \mu_s))$$

où $|\Sigma_s|$ est le déterminant de $\Sigma_s$.

Nous allons estimer la matrice $\hat{W}$ telle que l'ensemble des paramètres $\hat{\mathcal{M}}$ maximise la log-vraissemblance des observations grâce à un algorithme EM. Pour ce faire nous définissons la fonction auxiliaire

$$\mathcal{Q}(\tilde{\mathcal{M}}, \mathcal{M}) = \sum_{t=1}^{T} \sum_{\theta_t \in \mathcal{S}} \mathbb{P}(\theta_t|\mathcal{M}) \log(\mathbb{P}(O_t|\tilde{\mathcal{M}}, \theta_t))$$

On peut montrer que $\mathcal{Q}(\tilde{M}, \mathcal{M}) \geq 0$ implique $\mathcal{L}(\tilde{M}) \geq \mathcal{L}(\mathcal{M})$ (cf [13]). Nous allons donc utiliser l'algorithme suivant :

1. Estimer la matrice $\tilde{W}$ telle que $\tilde{W} = \texttt{argmax}_{\bar{W}} \mathcal{Q}(\bar{\mathcal{M}}, \mathcal{M})$
2. Mettre à jour les paramètres à l'aide de la matrice $\tilde{W}$.

3. Répéter ces opérations jusqu'à vérifier un critère de convergence.
4. Renvoyer la transformation $\hat{W}$ qui est la composition [1] des transformations $\tilde{W}$ estimées à chaque itération.

En faisant l'hypothèse que les matrices de covariance sont symétriques, on aboutit à une forme explicite pour le calcul de la matrice $\tilde{W}$ qui annule la différentielle de $\mathcal{Q}$, et qui est donc un extremum local. On peut calculer $W$ ligne par ligne itérativement grâce aux formules suivantes :

$$\tilde{w}_i^\top = G^{(i)-1} z_i^\top$$

où $\tilde{w}_i$ et $z_i$ sont respectivement les $i$-ième lignes de $\tilde{W}$ et $Z$, avec $G^{(i)} \in \mathcal{M}_{d+1,d+1}(\mathbb{R})$ telle que

$$g_{ij}^{(i)} = \sum_{s=1}^{S} v_{ii}^{(s)} d_{jq}^{(s)}$$

où les réels $v_{ii}^{(s)}$ et $d_{jq}^{(s)}$ sont respectivement les éléments des matrices $V^{(s)}$ et $D^{(s)}$ vérifiant

$$V^{(s)} = \sum_{t=1}^{T} \gamma_s(t) \Sigma_s^{-1}$$

et

$$D^{(s)} = \xi_s \xi_s^\top$$

avec $\xi_s = (\mu_s^\top, 1)^\top$.
On obtient donc les nouveaux paramètres $\tilde{\mathcal{M}}$ en posant

$$\tilde{\mathcal{M}} = \{\tilde{W}\xi_s, \Sigma_s, \alpha_s\}$$

On remet itérativement à jour les moyennes en recalculant une nouvelle matrice $\tilde{W}$ avec les nouveaux paramètres, jusqu'à respecter un critère d'arrêt.

**Critère d'arrêt :** Le critère d'arrêt est basé sur la variation de la log-vraissemblance au cours des itérations. Tout d'abord on calcul la différence de log-vraissemblance entre les anciens paramètres $\mathcal{M}$ et ceux mis à jour $\hat{\mathcal{M}}$. Si cette différence est inférieure à un seuil fixé, alors on dit que la condition de convergence est respectée. On définit une variable *hold* qui prend la valeur vraie si la condition de convergence est respectée. Si la condition de convergence est respectée pendant *min_hold* itérations, alors on arrête le programme et on renvoie la dernière matrice $\hat{W}$ estimée.

Cette variable *hold* a pour effet d'empêcher l'algorithme de s'arrêter trop tôt, lorsque la différence de log-vraissemblance est très faible entre 2 itérations, puis réaugmente à la 3-ème itération.

1. Le terme composition a du sens si on considère les transformations affines associées à ces matrices.

### 3.2.2 Moyennes et Variances

En reprenant les notations de la section précédente, nous pouvons aussi adapter les variances dans l'algorithme EM précédent. Pour cela on va estimer une matrice $\hat{H}$ en plus de la matrice $\hat{W}$ à chaque itération.

Nous voulons calculer un nouveaux jeu de paramètres $\hat{\mathcal{M}} = (\hat{\mu}_s, \hat{\Sigma}_s, \alpha_s)$ tel que

$$\hat{\mu}_s = \hat{W}\xi_s$$

et

$$\hat{\Sigma_s} = L_s \hat{H} L_s^\top$$

avec $L_s = B_s^{-1}$ et où $B_s$ est la matrice triangulaire inférieure dans la décomposition de cholesky de $\Sigma_s$, i.e.

$$\Sigma_s = B_s B_s^\top$$

Nous allons encore une fois utiliser un algorithme EM pour lequel nous allons utiliser la même fonction auxiliaire, mais cette fois-ci en deux étapes. Cet algorithme repose sur le principe suivant :
Notons $\tilde{\mathcal{M}} = (W\xi_s, \Sigma_s, \alpha_s)$ l'ensemble des paramètres avec les moyennes adaptées, et $\check{\mathcal{M}} = (\tilde{\mu}_s, L_s H L_s^\top, \alpha_s)$ celui avec moyennes et variances adaptées. L'algorithme est basé sur le fait que si $\mathcal{Q}(\tilde{\mathcal{M}}, \mathcal{M}) \geq 0$ et $\mathcal{Q}(\check{\mathcal{M}}, \tilde{\mathcal{M}}) \geq 0$ alors $\mathcal{L}(\check{\mathcal{M}}) \geq \mathcal{L}(\tilde{\mathcal{M}}) \geq \mathcal{L}(\mathcal{M})$.

1. Estimer la matrice $\tilde{W}$.
2. Mettre à jour les moyennes avec la matrice $\tilde{W}$.
3. Estimer la matrice $\tilde{H}$ en tenant compte de la modification des moyennes.
4. Mettre à jour les matrices de covariance.
5. Répéter les opérations jusqu'à vérifier un critère de convergence.
6. Renvoyer les transformations $\hat{W}$ et $\hat{H}$ qui sont la concaténation des transformations estimées à chaque itération.

Cet algorithme se place dans le cadre des algorithmes EM généralisés, où on optimise alternativement les moyennes puis les variances, au lieu de les optimiser de façon jointe en une seule étape. En supposant encore une fois que toutes les matrices de covariances sont diagonales, on arrive à une forme explicite pour l'estimation de la matrice $\hat{H}$ qui s'écrit comme suit

$$\hat{H} = \frac{\sum_{s=1}^{S} B_s^\top \left[\sum_{t=1}^{T} \gamma_s(t)(O_t - \hat{\mu}_s)(O_t - \hat{\mu}_s)^\top\right] Bs}{\sum_{s=1}^{S}\sum_{t=1}^{T} \gamma_s(t)}$$

avec $\gamma_s(t) = \mathbb{P}(\theta_t = s | \mathcal{M}, O_t)$ la probabilité que $O_t$ appartienne à la classe $\theta_t = s$.

Pour cet algorithme, le critère d'arrêt est le même que le précédent.

### 3.2.3 MLLR-Contraint

Dans le MLLR-Contraint, nous adaptons les moyennes et les variances en imposant à la transformation servant à adapter les matrices de covariances d'être égale a l'application linéaire associée à la transformation $\hat{W}$ servant à adapter les moyennes.

Dans ce cadre, nous avons $\hat{\mathcal{M}} = (\hat{\mu}_s, \hat{\Sigma}_s, \alpha_s)$ avec

$$\hat{\mu}_s = A\mu_s + b$$

et

$$\hat{\Sigma}_s = A\Sigma_s A^\top$$

avec $A \in \mathcal{M}_{d,d}(\mathbb{R})$ et $b \in \mathbb{R}^d$.

Cette adaptation des paramètres afin de maximiser la vraissemblance des données peut aussi être vu comme une adaptation des données afin de maximiser leur vraissemblances via la transformation $\tilde{O}_t = \tilde{A}O_t + b$ avec

$$\tilde{A} = A^{-1}$$

et

$$\tilde{b} = -A^{-1}b$$

Ainsi adapter les paramètres $\mathcal{M}$ en $\hat{\mathcal{M}}$ afin de maximiser la vraissemblance des données est équivalent à maximiser la vraissemblance des données $\tilde{O}$.

Afin de calculer la matrice $\hat{W} = [\tilde{A} \quad \tilde{b}]$, nous allons utiliser un algorithme EM. La procédure est là même que précedemment :

1. Estimer la matrice $\tilde{W}$ en fonction des paramètres courant.
2. Mettre à jour les observations grâce à la matrice $\tilde{W}$.
3. Répéter l'opération jusqu'à respecter un critère de convergence.

4. Renvoyer $\hat{W}$ qui est la concaténation des transformations estimées à chaque itération.

L'estimation de la matrice $\hat{W}$ à chaque étape se fait ligne par ligne itérativement, mais le système d'équation de l'estimation de chaque ligne dépend des autres. De ce fait, on ne calcule pas le maximum de la fonction auxiliaire à chaque étape, on ne fait qu'optimiser itérativement dans la direction donnée par la dérivée partielle par rapport à la ligne courante.

### 3.2.4 Données synthétiques

Comme expliqué dans le préambule, on ne peut théoriquement adapter correctement que $d+1$ classe dans un espace à dimension $d$. J'ai donc généré un mélange de trois gaussiennes dans un espace à deux dimensions. J'ai ensuite généré des données à partir d'un deuxième mélange de trois gaussiennes.

A l'aide de ce mélange de gaussienne et des données d'adaptation, j'ai appliqué une adaptation des moyennes seules. On remarque que l'algorithme se comporte particulièrement bien sur ce jeu de données synthétiques.

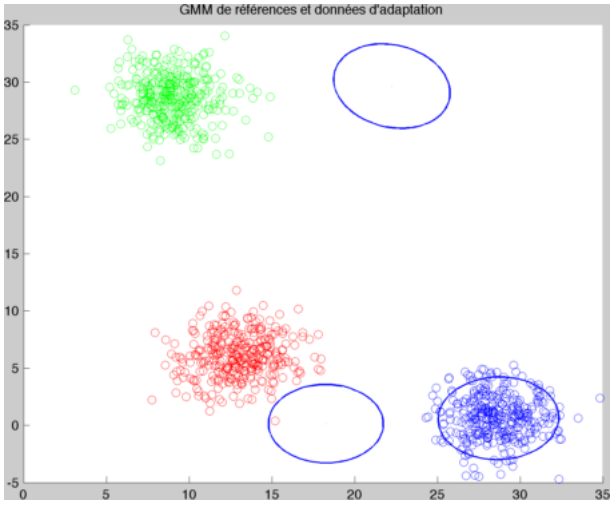



FIGURE 3.1 – GMM de référence et données d'adaptation.

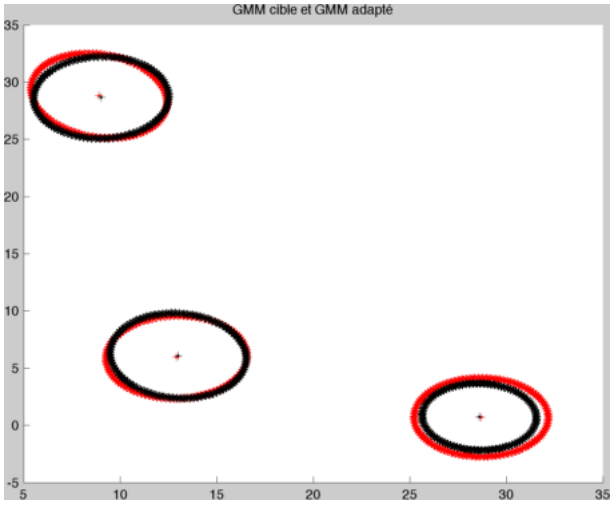



FIGURE 3.2 – GMM ciblé et GMM adapté. On remarque que l'adaptation est très performante.

Nous allons donc maintenant évaluer les performances des algorithmes à l'aide de notre de base de données de chant.

# Chapitre 4

# Evaluation des performances

Afin d'évaluer les performances de notre algorithme, nous avons mis en place des expériences pour tester l'adaptation. Nous décrivons d'abord ces expériences, puis nous évaluons les performances de notre algorithme selon deux critères : l'un quantitatif basé sur la théorie de l'information afin de mesurer l'information apportée par l'adaptation puis un critère plus qualitatif comparant l'évolution des taux de classification et les erreurs d'alignement entre les modèles appris hors-ligne et les modèles adaptés.

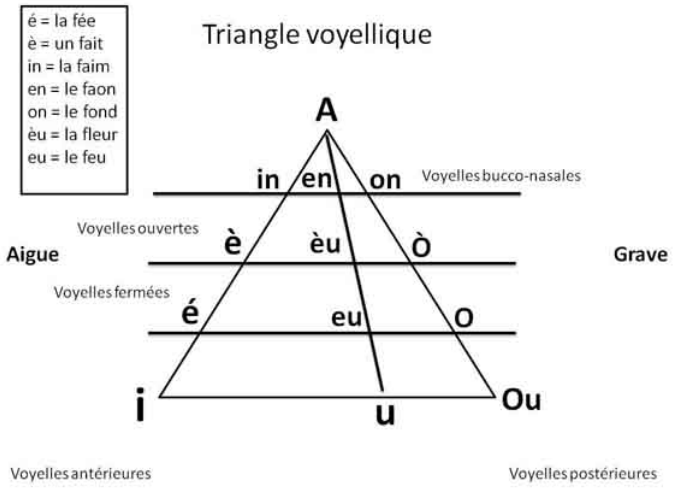
## 4.1 Protocole d'adaptation

### 4.1.1 Description

L'adaptation nécessite un modèle appris hors-ligne ainsi que des données d'adaptations. Il faut donc choisir des classes dont on veut adapter les probabilités d'observations, un locuteur de référence ainsi qu'un locuteur ciblé par l'adaptation.

**Les locuteurs** L'adaptation doit se faire d'un locuteur de référence vers un locuteur ciblé. Les paramètres du modèle d'observation du locuteur de référence sont adaptés pour maximiser la vraissemblance des données du locuteur ciblé.

**Les classes** On se fixe la liste de classe suivante : *' a ' , ' i ' , ' ou ' , ' eu ' , ' è ' , ' ò ' , ' u ' , 'èu'*.
Les trois premières classes correspondent au sommet du triangle vocalique, et sont donc normalement les classes les plus éloignées du point de vue de la perception sonore. Puis nous rajoutont la classe au centre du triangle vocalique, pour finir avec les milieux des côtés du triangle.
On réalise huit adaptations, en adaptant les paramètres des $k$ premières classes à la $k$-ième adaptation.




**Les données** Une fois les classes fixées, on choisit aléatoirement dans la base de données du locuteur ciblé des observations qui seront nos données d'adaptation. Nous choisissons entre 100 et 1000 données d'adaptation au total, chaque fois aléatoirement, et nous renvoyons le modèle adapté le plus vraissemblable. Cette approche est inspiré de l'algorithme RANSAC.

**Le modèle** Le modèle de référence est le modèle dont on a estimé les paramètres par apprentissage supervisé avec la base de donnée du locuteur de référence. Le modèle ciblé est le modèle correspondant aux données du locuteur ciblé. Nous voudrions que les paramètres du modèle de référence convergent vers ceux du modèle ciblé lors de l'adaptation.

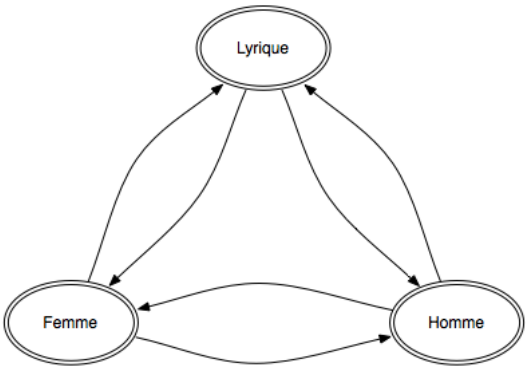
### 4.1.2 Adaptation GMM non-supervisée

En notant $\mathbf{o} = (o_t)_T$ les données d'adaptation et $\mathcal{M}$ les paramètres du modèle de référence, on applique les algorithmes d'adaptation de moyennes seules, moyennes et variances, puis MLLR-contraint afin d'estimer les nouveaux paramètres $\hat{\mathcal{M}}$ qui maximisent la vraissemblance de $\mathbf{o}$.

Le problème que l'on peut rencontrer à partir de l'adaptation simultanée de deux classes, est que l'algorithme peut maximiser la vraissemblance des données d'adaptation en permutant les classes 1 et 2. Ainsi, en terme de classification nous aurons un taux extrêmement faible alors que l'adaptation se sera plutôt bien déroulée. Afin d'éviter ce problème, nous avons évalué les performances de classification à une permutation près après avoir adapté notre modèle.
La permutation choisie pour évaluer la performance en terme de classification est celle qui maximise le taux de bonne classification i.e. celle pour laquelle la classification par GMM adapté et permuté se rapproche le plus de la vérité terrain.

## 4.2 Amélioration qualitative

Etant donné que nous avions trois corpus de voix chantée à notre disposition, nous avons effectué six adaptations, correpondant à chacune des flèches du graphe suivant :




Nous allons donc pouvoir analyser le comportement des différents algorithmes avec les résultats suivants.

### 4.2.1 Gains en classification

Afin d'analyser l'amélioration apportée par l'adaptation dans un cadre de classification de phonème, j'ai rempli les tableaux montrant les différents scores de chacun des modèles, avant et après adaptation. Chaque colonne correspond au pourcentage de bonne classification par le GMM représentant cette colonne.

**#classes** : Nombre de classe adaptées.

**Cible** : Modèle cible.

**Référence** : Modèle de référence.

**Moyennes** : Adaptation des moyennes seules.

**Variances** : Adaptation des moyennes et des variances.

**Contraint** : Adaptation des moyennes et des variances contraintes.

*Ex : dans une adaptation homme vers femme, le locuteur de référence est l'homme tandis que le locuteur ciblé est la femme.*

| Adaptation homme vers femme | | | | | |
|---|---|---|---|---|---|
| #classes | Cible | Référence | Moyennes | Variances | Contraint |
| 1 | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| 2 | 99.8% | 85.5% | 99.6% | 99.7% | 99.8% |
| 3 | 99.4% | 77.2% | 99.3% | 99.6% | 89.3% |
| 4 | 98.3% | 65.9% | 73.4% | 72.7% | 87.3% |
| 5 | 96.2% | 57.1% | 69.3% | 69.8% | 75.3% |
| 6 | 92.6% | 52.2% | 59.2% | 60.5% | 67.3% |
| 7 | 92.7% | 51.0% | 53.5% | 62.3% | 68.5% |
| 8 | 90.9% | 46.6% | 56.6% | 59.1% | 68.3% |

| Adaptation femme vers homme | | | | | |
|---|---|---|---|---|---|
| #classes | Cible | Référence | Moyennes | Variances | Contraint |
| 1 | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| 2 | 100.0% | 99.7% | 100.0% | 99.9% | 99.9% |
| 3 | 99.7% | 38.9% | 99.1% | 99.4% | 91.0% |
| 4 | 99.2% | 32.0% | 98.1% | 74.4% | 81.3% |
| 5 | 98.7% | 28.0% | 52.6% | 50.0% | 71.1% |
| 6 | 98.0% | 25.9% | 79.4% | 76.6% | 65.7% |
| 7 | 97.8% | 8.9% | 73.9% | 74.2% | 62.8% |
| 8 | 97.1% | 8.3% | 59.9% | 63.6% | 59.9% |

| Adaptation homme vers lyrique | | | | | |
|---|---|---|---|---|---|
| #classes | Cible | Référence | Moyennes | Variances | Contraint |
| 1 | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| 2 | 99.6% | 69.0% | 99.3% | 99.3% | 55.6% |
| 3 | 99.3% | 70.2% | 95.9% | 98.8% | 59.7% |
| 4 | 97.5% | 58.7% | 70.2% | 90.6% | 77.4% |
| 5 | 95.9% | 51.4% | 58.4% | 90.5% | 75.2% |
| 6 | 93.1% | 47.1% | 51.8% | 74.8% | 62.1% |
| 7 | 92.2% | 38.2% | 53.1% | 57.4% | 59.4% |
| 8 | 87.6% | 35.7% | 47.9% | 56.9% | 63.6% |

| Adaptation lyrique vers homme | | | | | |
|---|---|---|---|---|---|
| #classes | Cible | Référence | Moyennes | Variances | Contraint |
| 1 | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| 2 | 100.0% | 56.6% | 100.0% | 100.0% | 56.3% |
| 3 | 99.7% | 51.7% | 99.7% | 99.7% | 51.4% |
| 4 | 99.2% | 53.1% | 98.8% | 78.0% | 88.0% |
| 5 | 98.7% | 46.5% | 97.8% | 98.4% | 67.0% |
| 6 | 98.0% | 43.0% | 73.6% | 97.8% | 79.2% |
| 7 | 97.8% | 39.4% | 77.0% | 78.5% | 70.6% |
| 8 | 97.1% | 37.0% | 75.9% | 95.6% | 39.7% |

| Adaptation femme vers lyrique | | | | | |
|---|---|---|---|---|---|
| #classes | Cible | Référence | Moyennes | Variances | Contraint |
| 1 | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| 2 | 99.6% | 68.9% | 99.0% | 99.2% | 55.6% |
| 3 | 99.3% | 57.1% | 93.0% | 96.9% | 88.7% |
| 4 | 97.5% | 49.0% | 72.2% | 69.9% | 82.7% |
| 5 | 95.9% | 43.1% | 46.8% | 57.9% | 37.4% |
| 6 | 93.1% | 20.6% | 49.4% | 59.4% | 49.4% |
| 7 | 92.2% | 19.6% | 45.0% | 55.6% | 55.1% |
| 8 | 87.6% | 18.0% | 55.8% | 59.6% | 52.5% |

| Adaptation lyrique vers femme | | | | | |
|---|---|---|---|---|---|
| #classes | Cible | Référence | Moyennes | Variances | Contraint |
| 1 | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| 2 | 99.8% | 99.6% | 99.7% | 99.7% | 99.8% |
| 3 | 99.4% | 89.8% | 99.6% | 99.6% | 89.3% |
| 4 | 98.3% | 79.8% | 96.8% | 96.8% | 87.4% |
| 5 | 96.2% | 58.6% | 93.2% | 93.9% | 83.6% |
| 6 | 92.6% | 53.5% | 83.1% | 90.1% | 75.9% |
| 7 | 92.7% | 46.4% | 76.1% | 84.8% | 76.6% |
| 8 | 90.9% | 42.7% | 72.7% | 66.9% | 81.7% |

### 4.2.2 Discussion des résultats

La première remarque est que lorsque le nombre de classe augmente, les taux de bonnes classifications du modèle de référence diminuent rapidement. Cela confirme le besoin d'adaptation du modèle mis en avant dans la section 2. En réalisant une adaptation, on remarque qu'en générale on augmente le taux de classification, sauf dans certains cas pour le MLLR-Contraint. Ceci est dû à un comportement particulier que nous détaillerons plus tard.

Contrairement à ce que l'on pourrait attendre, nous n'avons pas systématiquement un meilleur score en adaptant moyennes et variances qu'en adaptant seulement les moyennes. Cela est dû au fait que l'adaptation étant non supervisée, l'algorithme ne sait pas correctement adapter les différentes gaussiennes une par une.

On remarque sur les figures que lors de l'adaptation des variances, l'algorithme a placé deux gaussiennes à gauche, alors qu'il n'y avait qu'une seule classe. Mais cela ne semble

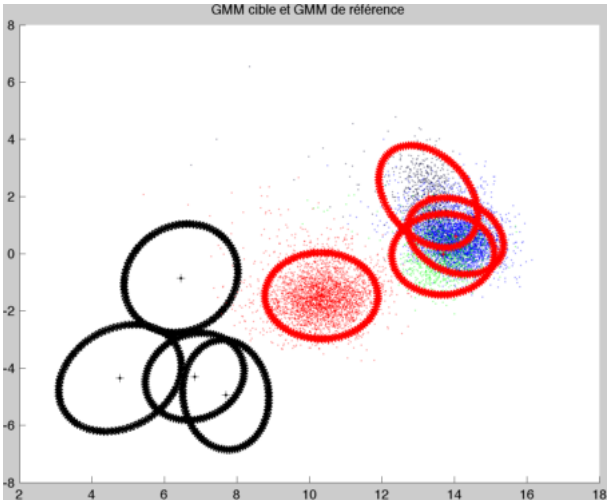



FIGURE 4.1 – Modèle cible, modèle de référence et données d'adaptation

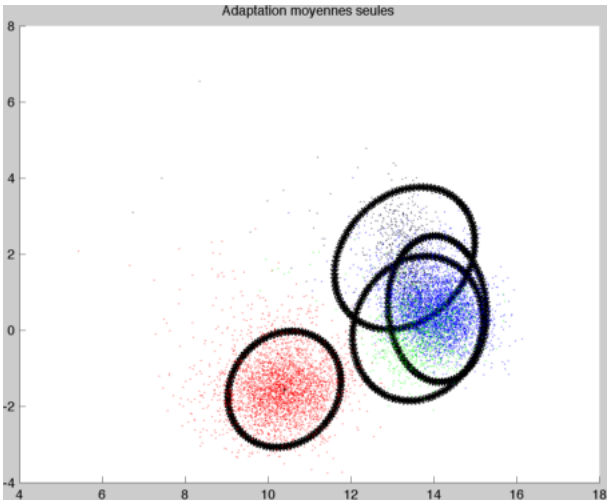



FIGURE 4.2 – Adaptation des moyennes seules.

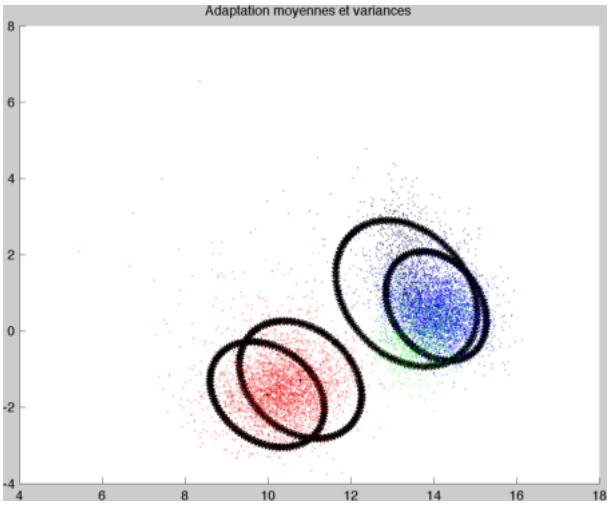



FIGURE 4.3 – Adaptation des moyennes et des variances.

pas très aberrant si on oublie l'étiquetage des données.

Autrement dit le modèle adapté le plus vraissemblable n'est pas celui qui maximise la le taux de bonne classification et le modèle effectivement ciblé n'est pas celui issue de notre apprentissage supervisé. En permettant aux covariances de s'adapter nous amplifions ce phénomène, dégradant d'autant plus le taux de bonne classification. Ce problème a donc deux possibilités d'améliorations : donner un meilleur apriori à notre modèle, ou alors modifier la fonction objectif.

## 4.3 Gains en terme d'information

Nous avons calculé la distance de Kullback-Leibler entre les modèles de référence et les modèles ciblés, ainsi on remarque qu'il y a un net gain d'information apporté par l'adaptation des moyennes, et un gain supplémentaire mais moindre apporté par l'adaptation des moyennes. La distribution du modèle adapté se rapproche donc considérablement de la distribution du modèle ciblé. Une première chose à remarquer est que la distance de Kullback-Leibler ne dépend pas des labels de chaque classe, i.e. elle ne dépend pas de la permutation calculée lors de l'évaluation des performances en terme de classification.

Comparée aux distances entre les distributions, les taux de classifications n'augmentent pas beaucoup et ceci met en exergue l'importance de supervision lors de l'apprentissage. Le même phénomène se produit lorsque l'on apprend les paramètres du GMM avec l'algorithme EM classique non-supervisé. Ce sont les limites imposées par les hypothèses sous-jacentes au modèle, à savoir que les distributions conditionnelles des observations conditionnellement à une classe sont des gaussiennes, et que les observation sont i.i.d (indépendante et identiquement distribuée).

## 4.4 Le MLLR-contraint

L'algorithme du MLLR-contraint met parfaitement en avant le fait que maximiser la vraissemblance des données est complètement indépendant de l'amélioration du taux de classification. Il choisit généralement d'augmenter la (ou les) variance de la gaussienne la plus proche du paquet de données, et de la translater afin que toutes ces données appartiennent à cette classe. Ceci ne peut pas se produire dans l'adaptation des moyennes seules, car la vraissemblance d'un modèle où une classe tenterait d'englober toutes les données serait trop faible. Lorsque l'on effectue l'adaptation des moyennes non contraintes, on fait une première adaptation des moyennes à la première itération juste avant d'adapter les covariances, ainsi nous avons un apriori bien mieux distribué sur l'ensemble des classes à adapter. Ainsi, l'adaptation des matrices de covariance reste raisonnable.

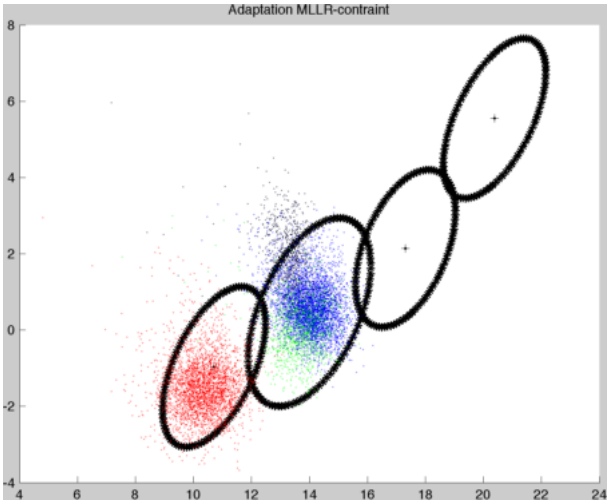



FIGURE 4.4 – Adaptation MLLR-contraint où la vraissemblance des données est supportée par 2 classes.

La justification mathématique de phénomène est que lorsque l'on s'éloigne de l'enveloppe convexe de l'ensemble des moyennes du modèle, la probabilité à posteriori que l'observation appartienne à la classe la plus proche d'elle tend vers 1, tandis que les autres tendent vers 0. Deux solutions peuvent être proposées pour améliorer ces résultats :

— On fournit un meilleur apriori à notre algorithme, afin de le forcer à distribuer l'adaptation.
— On rajoute un degré de liberté au modèle d'adaptation, ce qui implique de modifier la fonction objectif.

# Bibliographie


[1] Arshia Cont. A coupled duration-focused architecture for real-time music-to-score alignment. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(6) :974–987, 2010.

[2] Stephen Cox. Predictive speaker adaptation in speech recognition. *Computer Speech & Language*, 9(1) :1–17, 1995.

[3] Sadaoki Furui. A training procedure for isolated word recognition systems. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(2) :129–136, 1980.

[4] Mark JF Gales. Maximum likelihood linear transformations for hmm-based speech recognition. *Computer speech & language*, 12(2) :75–98, 1998.

[5] Mark JF Gales and Philip C Woodland. Mean and variance adaptation within the mllr framework. *Computer Speech & Language*, 10(4) :249–264, 1996.

[6] Rong Gong, Philippe Cuvillier, Nicolas Obin, and Arshia Cont. Real-time audio-to-score alignment of singing voice based on melody and lyric information. In *Interspeech*, 2015.

[7] Denny Iskandar, Ye Wang, Min-Yen Kan, and Haizhou Li. Syllabic level automatic synchronization of music signals and text lyrics. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 659–662. ACM, 2006.

[8] Christopher J Leggetter and Philip C Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech & Language*, 9(2) :171–185, 1995.

[9] Alex Loscos, Pedro Cano, and Jordi Bonada. Low-delay singing voice alignment to text. In *Proceedings of the ICMC*, volume 18. Citeseer, 1999.

[10] Namunu C Maddage, Khe Chai Sim, and Haizhou Li. Word level automatic alignment of music and lyrics using vocal synthesis. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 6(3) :19, 2010.

[11] Oscar Mayor, Jordi Bonada, and Alex Loscos. The singing tutor : Expression categorization and segmentation of the singing voice. In *Proceedings of the AES 121st Convention*, 2006.

[12] Annamaria Mesaros and Tuomas Virtanen. Automatic alignment of music audio and lyrics. In *Proceedings of the 11th Int. Conference on Digital Audio Effects (DAFx-08)*, 2008.

[13] Alexis Roche. Em algorithm and variants : An informal tutorial. *arXiv preprint arXiv :1105.1476*, 2011.

[14] Koichi Shinoda and Chin-Hui Lee. Structural map speaker adaptation using hierarchical priors. In *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 381–388. IEEE, 1997.

[15] Phil C Woodland. Speaker adaptation for continuous density hmms : A review. In *ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition*, 2001.