Nouvelle approche pour le regroupement des locuteurs dans des émissions radiophoniques et télévisuelles

Mickael Rouvier Sylvain Meignier
LIUM, Université du Maine, France
{mickael.rouvier, sylvain.meignier}@lium.univ-lemans.fr

RÉSUMÉ

Dans cet article, nous proposons un nouveau modèle de regroupement de locuteurs pour la tâche de segmentation et de regroupement de locuteurs. Un des problèmes majeur rencontré dans le regroupement des locuteurs est que les algorithmes d'agglomération hiérarchique utilisés ne garantissent pas de donner une solution optimale. Nous proposons d'exprimer le problème de regroupement des locuteurs comme un problème de Programmation Linéaire en Nombre Entier (PLNE). Ainsi, un solveur PLNE peut être utilisé lequel ira chercher la solution optimale de regroupement de locuteurs sur l'ensemble du problème. Les expériences ont été conduites sur le corpus journalistique français ESTER-2. Avec ce nouveau modèle de regroupement de locuteurs, le DER décroît de 2,43 points absolus.

ABSTRACT

New approach for speaker clustering of broadcast news

In this paper, we propose a new clustering model for speaker diarization. A major problem with using greedy agglomerative hierarchical clustering for speaker diarization is that they do not guarantee an optimal solution. We propose a new clustering model, by redefining clustering as a problem of Integer Linear Programming (ILP). Thus an ILP solver can be used which searches the solution of speaker clustering over the whole problem. The experiments were conducted on the corpus of French broadcast news ESTER-2. With this new clustering, the DER decreases by 2.43 points.

MOTS-CLÉS : segmentation et regroupement de locuteur, programmation linéaire en nombres entiers, i-vecteur.

KEYWORDS: speaker diarization, integer linear programming, i-vector.

1 Introduction

L'objectif de la Segmentation et du Regroupement de Locuteurs (SRL) consiste à découper en tour de parole un enregistrement audio et à regrouper les zones dès lors qu'elles appartiennent à un même locuteur afin de répondre à la question : "qui parle et quand ?". Cette opération est réalisée sans information *a priori* ni sur le nombre de locuteurs, ni sur leur identité. L'approche classique consiste à découper le signal audio en segments et à les regrouper dans des classes, où chaque classe contient les segments d'un seul et même locuteur.

Actuellement, les principales méthodes de regroupement en locuteurs sont basées sur des algorithmes d'agglomération hiérarchique gloutonne tels que les algorithmes : ascendant (Barras *et al.*, 2006) ou descendant (Fredouille et Senay, 2006). Les systèmes ayant une approche ascendante (connus aussi sous le nom de Regroupement Agglomératif Hiérarchique (RAH)) ont obtenu les

Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 1: JEP, pages 97–104,
Grenoble, 4 au 8 juin 2012.

meilleurs résultats lors des évaluations ESTER et NIST. Le RAH est un algorithme itératif qui cherche à chaque itération à agglomérer les deux classes les plus similaires. Ce processus est itéré tant que la similarité entre les 2 classes les plus proches soit inférieure à un seuil fixé. Cette similarité est calculée à partir des vraisemblances obtenues via des Modèles de Mélanges de Gaussiennes (GMM). Malheureusement, les algorithmes gloutons basés sur les GMM souffrent de deux principaux inconvénients.

Le premier étant que les approches gloutonnes sont des algorithmes itératifs qui vont, à chaque itération, prendre une décision localement optimale dans l'espoir de proposer un résultat globalement optimal. Cependant, durant cette recherche, la sélection des deux prochaines classes à regrouper dépend fortement de celles choisies précédemment. Un mauvais regroupement n'est jamais remis en cause et il est conservé jusqu'à la fin pouvant causer une augmentation du nombre d'erreurs.

Deuxièmement, le regroupement des locuteurs se fait à partir de GMM appris sur le signal audio. Malheureusement, le signal audio ne véhicule pas seulement l'information sur les locuteurs (l'information utile) mais aussi d'autres informations qui peuvent venir perturber le processus de regroupement des locuteurs. Ces informations inutiles peuvent être de différentes natures et peuvent être liées à la variabilité de l'environnement (environnement bruité...), la variabilité du canal (microphone, téléphone...), la variabilité du locuteur (émotion...), etc...

Dans cet article, nous proposons un nouveau modèle de regroupement de locuteurs où, contrairement aux approches gloutonnes le processus de regroupement des locuteurs se fait de manière globale sur l'ensemble du problème. Nous proposons de remplacer la recherche gloutonne par une formulation optimale. En donnant quelques définitions générales sur les classes, l'algorithme ascendant peut être exprimé sous forme de problème de Programmation Linéaire en Nombre Entier (PLNE). Ainsi un solveur PLNE peut être utilisé pour minimiser le résultat de la fonction objective, lequel va chercher la solution optimale de regroupement des locuteurs sur l'ensemble du problème. Ce nouveau modèle PLNE est basé sur les i-vecteurs, une technique introduite dans le domaine de la vérification qui permet de modéliser uniquement l'information du locuteur.

Cet article est organisé comme suit. La Section 2 présente tout d'abord l'architecture du système, puis la Section 3 le corpus utilisé. Ensuite, l'approche des i-vecteurs est expliquée dans la Section 4. La Section 5 présente notre cadre de travail pour le regroupement de locuteurs ainsi que les résultats de nos expériences. Nos conclusions sont résumées dans la dernière partie (Section 6).

2 Architecture du système

Le système utilisé est celui du LIUM Speaker Diarization(Meignier et Merlin, 2010), disponible sous licence GPL [1]. Ce système a obtenu les meilleurs résultats durant la campagne d'évaluation ESTER-2.

Le système est composé d'une segmentation acoustique basée sur le BIC (*Bayesian Information Criterion*) suivi par un regroupement hiérarchique lui aussi basé sur le BIC. Chaque classe représente un locuteur et est modélisée avec une Gaussienne de covariance pleine. Un décodage en Viterbi est utilisé pour ajuster les frontières des segments en utilisant un GMM avec 8 composantes diagonales. Le regroupement de locuteurs est réalisé sur une paramétrisation acoustique de 12 MFCC+E, calculée sur une fenêtre de 10ms. La musique et les jingles sont supprimés en utilisant un décodage Viterbi avec 8 GMMs.

1. http ://www-lium.univ-lemans.fr/diarization/

Lors de ces étapes, l'environnement sonore aide le système à détecter les locuteurs : les paramètres ne sont donc pas normalisés. Parfois un locuteur est représenté par plusieurs classes qui contiennent les interventions de celui-ci en fonction de l'environnement sonore (bruit, musique, calme...). La contribution de l'environnement sonore doit alors être réduite et normalisée afin de regrouper ces classes en une seule.

La méthode classique consiste à faire un regroupement hiérarchique ascendant. Il est donc effectué sur les classes obtenues après la segmentation Viterbi : les paramètres de chaque segment sont normalisés et un modèle du monde est adapté (MAP) pour chaque classe. A chaque itération sont regroupées les 2 classes qui maximisent le critère NCLR (*Normalized Cross Likelihood Ratio*) (Le *et al.*, 2007). Le regroupement s'arrête lorsque la valeur de NCLR dépasse un seuil.

Dans cet article, nous proposons une autre méthode de regroupement des classes basée sur les i-vecteurs. Il s'agit juste de remplacer la dernière brique de regroupement des classes, le NCLR, par notre modèle. Tout le reste du processus de SRL, paramétrisation du signal audio, segmentation et regroupement BIC, reste valable.

3 Corpus

Les données utilisées pour les expériences sont celles de la campagne d'évaluation d'ESTER-2 (Galliano *et al.*, 2009). Elles sont composées d'émissions enregistrées sur 4 radios journalistiques françaises. Les données sont divisées en trois corpus : le corpus d'apprentissage correspondant à 111 émissions (90 heures de données), le corpus de développement correspondant à 20 émissions, et le corpus d'évaluation qui contient 26 émissions. Le corpus d'entraînement a été utilisé pour apprendre et conditionner les i-vecteurs et le corpus de développement pour choisir les différents paramètres de chaque système.

4 I-vecteur

4.1 Extraction des i-vecteurs

Dans le domaine de la vérification du locuteur, les i-vecteurs sont devenus état de l'art. Ils fournissent un cadre de travail élégant, permettant de réduire la taille d'un vecteur de très grande dimension en un vecteur plus compact, où toute l'information importante du locuteur est conservée. La technique est issue du cadre de travail Joint Factor Analysis (JFA), qui a été introduit dans (Kenny *et al.*, 2007). Ainsi, pour un GMM dépendant du locuteur et du canal où M est un super-vecteur correspondant aux moyennes du GMM, les i-vecteurs peuvent être exprimés comme suit :

$$M = m + Tw \tag{1}$$

où m est le super-vecteur correspondant aux moyennes concaténées d'un Modèle Universel (Universal Background Model - UBM) ; T est une matrice rectangulaire couvrant l'ensemble des variabilités importantes du locuteur ; w est un vecteur compact distribué selon $\mathbf{N}(0, \mathbf{I})$.

Plusieurs itérations sont nécessaires pour estimer la matrice T sur le corpus d'apprentissage, l'Equation 1 permet d'utiliser un vecteur compact w comme un modèle de locuteur en remplacement du

GMM. w est nommé par la suite i-vecteur. L'algorithme des i-vecteurs est décrit plus longuement dans (Dehak *et al.*, 2010).

4.2 Conditionnement des i-vecteurs

A cette étape les i-vecteurs contiennent l'information liée aux locuteurs mais aussi l'information inutile (canal, environnement...). Dans (Bousquet *et al.*, 2011), l'auteur propose une méthode robuste de conditionnement des i-vecteurs afin de modéliser cette information inutile. Cette méthode est un processus itératif qui a 2 buts :

1) S'assurer que les i-vecteurs sont distribués selon la loi $\mathbf{N}(0, \mathbf{I})$. Une des conséquences de cette contrainte est que les i-vecteurs deviennent ainsi indépendants.

2) Normaliser les i-vecteurs par leur longueur. Dans (Bousquet *et al.*, 2011; Garcia-Romero et Espy-Wilson, 2011), il a été montré que cela contribue à rendre gaussiennes les données et à rapprocher le corpus d'apprentissage et le corpus de test.

Dans le corpus d'apprentissage, pour chaque tour de parole obtenu en utilisant la segmentation en locuteur de référence nous calculons les i-vecteurs. L'algorithme de conditionnement consiste à extraire sur les i-vecteurs du corpus d'apprentissage, des paramètres de conditionnement et de les appliquer sur les i-vecteurs extraits du corpus de test.

L'Algorithme 1 décrit la méthode d'apprentissage des paramètres pour le conditionnement des i-vecteurs. Les paramètres (la moyenne μ_i et la matrice de covariance Σ_i) des i-vecteurs calculés sur le corpus d'apprentissage sont sauvegardés à chaque itération i (étape 0). Puis, les i-vecteurs sont conditionnés en utilisant les paramètres de l'itération actuelle. Ainsi, l'étape 1 consiste à centrer-réduire les i-vecteurs, et l'étape 2 à normaliser les i-vecteurs par leur longueur.

Algorithm 1: Algorithme de conditionnement des i-vecteurs sur le corpus d'apprentissage

for $i = 1$ *a* $nb\_iterations$ **do**
  Etape 0 : Calculer la moyenne μ_i et la matrice de covariance Σ_i sur le corpus d'apprentissage;
  for *chaque w dans le corpus d'apprentissage :* **do**
    Etape 1 : $w = \Sigma_i^{-\frac{1}{2}} (w - \mu_i)$;
    Etape 2 : $w = \frac{w}{\|w\|}$;
  end
end

Sur notre corpus de test, après le processus de regroupement des locuteurs donné par le BIC, un i-vecteur est calculé pour chaque classe. Ces i-vecteurs sont conditionnés itérativement en appliquant l'algorithme 2. L'algorithme 2 est proche de l'algorithme 1. Les différences sont situées dans l'absence de l'étape 0 : la moyenne μ_i et la matrice de covariance Σ_i utilisées pour chaque itération i sont celles sauvegardées durant la phase d'apprentissage.

4.3 La distance

Pour deux i-vecteurs w_i et w_j, le but est de vérifier s'ils correspondent au même locuteur. Si nous assumons l'homoscédasticité (égalité des variances), alors la distance entre deux i-vecteurs peut

Algorithm 2: Algorithme de conditionnement des i-vecteurs pour la phase de test

for $i = 1$ *a* $nb\_iterations$ **do**

Etape 1 : $w = \Sigma_i^{-\frac{1}{2}} (w - \mu_i)$;

Etape 2 : $w = \frac{w}{\|w\|}$;

end

s'écrire ainsi :

$$d(w_i, w_j) = \left(w_i - w_j\right) W^{-1} \left(w_i - w_j\right)' \tag{2}$$

où W est une matrice de covariance intra-classe calculée sur les i-vecteurs du corpus d'apprentissage conditionnés. Cette matrice de covariance intra-classe est calculée comme suit :

$$W = \frac{1}{n} \sum_{s=1}^{S} \sum_{i=1}^{n_s} \left(w_i^s - \bar{w^s}\right) \left(w_i^s - \bar{w^s}\right)' \tag{3}$$

où n_s est le nombre de segments pour un locuteur s, n est le nombre total de segments, w_i^s est un i-vecteur du corpus d'apprentissage du locuteur s pour une session i et $\bar{w}_s$ est la moyenne des i-vecteurs du locuteur s.

5 Nouveau modèle de regroupement de locuteurs global

Les décisions prises à chaque itération par les algorithmes RAH, ne garantissent pas de donner une solution optimale. Nous proposons d'écrire notre problème de regroupement de locuteurs sous forme de PLNE (Programmation Linéaire en Nombre Entier).

Le problème de regroupement peut être décrit de la façon suivante. Étant donnée une segmentation initiale (donné par le BIC), un i-vecteur est extrait pour chaque classe. Notre but est de regrouper les N i-vecteurs dans K classe (où K est à déterminer et est compris entre 1 et N). Pour transformer notre problème sous forme de PLNE, nous prenons comme hypothèse qu'une classe k est dotée d'un centre et qu'un i-vecteur peut appartenir à la classe si sa distance entre le centre de la classe k et le i-vecteur n est inférieure à une distance fixée *a priori*. Le centre d'une classe est obligatoirement un i-vecteur issu de notre problème. Théoriquement, il peut y avoir autant de classes que de i-vecteurs. Le but est de minimiser le nombre de classes k, de telle sorte que tous les i-vecteurs soient attribués à une classe et qu'un i-vecteur appartienne à une et une seule classe.

A partir de ces descriptions, nous pouvons formuler les contraintes et la fonction objective de notre problème. La fonction objective consiste à minimiser le nombre de classes k, mais aussi de minimiser la dispersion des i-vecteurs pour l'ensemble des classes. Nous définissons deux variables binaires y_k et $x_{k,n}$. La variable binaire y_k permet de savoir si la classe k est sélectionnée. La variable binaire $x_{k,n}$ permet de savoir si le i-vecteur n appartient à la classe k. Ainsi notre fonction objective peut s'écrire :

$$z = \sum_{k=1}^{N} y_k + \frac{1}{F} \sum_{k=1}^{N} \sum_{n=1}^{N} d(w_k, w_n) x_{k,n} \tag{4}$$

La fonction objective se compose de deux parties : la première ($\sum_{k=1}^{K} y_k$) calcule le nombre de classes présentes dans notre problème ; la seconde ($\sum_{k=1}^{K}\sum_{n=1}^{N} d(w_k, w_n)x_{k,n}$) calcule la somme des distances entre les centres des k classes et leurs i-vecteurs. Où $d(w_k, w_n)$ correspond à la distance entre le centre de la classe k et un i-vecteur n. La résolution de notre problème cherche à minimiser le nombre de classes et la dispersion des classes avec F un facteur de normalisation, permettant de pondérer les deux sous-parties de l'équation 4.

Nous rappelons, d'après nos hypothèses, que le centre de la classe est en réalité un i-vecteur (un segment) et que le calcul de la distance entre le centre de la classe k et le i-vecteur n n'est en réalité que le calcul de la distance entre le i-vecteur k et le i-vecteur n.

Notre nouveau modèle de regroupement de locuteurs s'écrit donc :

$$
\begin{array}{lll}
\text{Minimize} & z & \\
\text{Subject To} & \sum_{n=1}^{N} x_{k,n} = 1, & \forall k, (5) \\
& x_{k,n} - y_k \leq 0, & \forall k, \forall n, (6) \\
& d(w_k, w_n)x_{k,n} \leq \delta, & \forall k, \forall n, (7) \\
& x_{k,n} \in \{0,1\}, & \forall k, \forall n \\
& y_k \in \{0,1\}, & \forall k
\end{array}
$$

Nous nous assurons, dans l'équation 5, que l'ensemble des i-vecteurs ait été assigné à une seule classe. Dans l'équation 6, nous nous assurons que si un i-vecteur n est attribué à une classe k, alors la classe k est sélectionnée. Dans l'équation 7, un segment n peut être sélectionné dans une classe k si sa distance est plus petite ou égale à une distance δ. $d(w_k, w_n)$ correspond à la distance donnée par l'Equation 2 entre le i-vecteur n et la classe k.

6 Résultat et comparaison

6.1 I-Vecteur et PLNE

La matrice T de l'Equation 1 est estimée sur le corpus d'apprentissage. La matrice est itérativement estimée utilisant l'algorithme d'espérance-maximisation (EM). Nous utilisons une paramétrisation acoustique de dimension 60 : composée de 19 MFCC plus l'énergie complétée de la dérivée première et seconde. Le modèle du monde GMM-UBM est un modèle indépendant du genre et du canal. Il est composé de 1024 Gaussiennes et est appris en utilisant l'outil Alize [2].

Afin d'avoir un équilibre entre la précision du modèle et la quantité de données menant à l'estimation des paramètres, nous avons choisi de fixer la dimension des i-vecteurs à 60. En effet, si nous choisissons une dimension de i-vecteurs supérieure pour des segments ayant une durée trop courte, la matrice T ne peut pas être correctement estimée.

Le solveur de PLNE utilisé est celui fourni gratuitement sous Linux : GNU Linear Programming Kit [3].

2. http ://alize.univ-avignon.fr/
3. http ://www.gnu.org/s/glpk/

6.2 Résultats

Dans un premier temps, nous cherchons à déterminer la distance à utiliser dans le modèle de PLNE (Equation 7). Dans la Figure 1, nous faisons varier la distance utilisée dans le modèle de PLNE (axe des abscisses) et reportons le DER (*Diarization Error Rate*) obtenu (axe des ordonnées) et ceci sur les corpus de développement et de test.

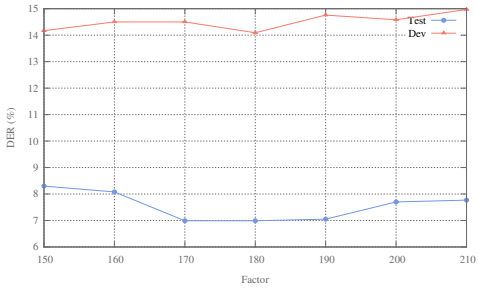

FIGURE 1 – DER en fonction de la distance utilisée dans le modèle PLNE

Nous observons dans la Figure 1 que la distance qui minimise le DER sur le corpus de développement est à 180. Celle-ci est exactement la même que sur le corpus de test.

Dans le Tableau 1, nous proposons pour l'algorithme de RAH de comparer les i-vecteurs (*i-vecteur RAH*) par rapport au NCLR (*NCLR RAH*). Puis nous proposons de voir l'apport du modèle de regroupement global sur le système à base de i-vecteurs (*i-vecteur PLNE*). Le système *NCLR RAH* est le système classique utilisé pendant la campagne d'évaluation ESTER-2. La différence entre ces 3 systèmes se situe uniquement sur le remplacement de la dernière brique NCLR.

TABLE 1 – Méthode de regroupement des classes (DER sur le corpus d'évaluation)
NCLR RAH : le système classique
i-vecteur RAH : le système utilisant les i-vecteurs et l'algorithme ascendant
i-vecteur PLNE : le système utilisant les i-vecteurs et le modèle de PLNE

Corpus	NCLR RAH	i-vecteur RAH	i-vecteur PLNE
Africa 1	9,60%	6,05%	2,79%
Inter	9,23%	11,72%	8,62%
RFI	3,61%	2,33%	2,33%
TVME	13,31%	13,17%	13,54%
ESTER-2	9,42%	9,08%	6,99%

Nous constatons une réduction du DER de 0,34 point absolu entre les systèmes *NCLR RAH* et *i-vecteur RAH*. Le remplacement par un modèle de regroupement global sur les i-vecteurs *i-vecteur PLNE* permet une réduction du DER d'environ 2 points absolus par rapport au système *i-vecteur RAH* et de 2,43 points absolus par rapport au système *NCLR RAH*. On observe, toujours sur le système *i-vecteur PLNE*, une réduction du DER sur l'ensemble des émissions, sauf pour les émissions TVME. En effet, sur les émissions TVME la plupart des locuteurs (56% des locuteurs) interviennent

en utilisant un téléphone, ce qui peut poser un problème puisque l'extraction des i-vecteurs se fait à partir d'un GMM-UBM indépendant du canal et du genre.

7 Conclusion

Dans cet article, nous avons proposé un nouveau modèle de regroupement des locuteurs basé sur les i-vecteurs. Dans le processus de SRL, la dernière brique du regroupement de locuteurs (le NCLR) a été remplacée par notre nouveau modèle, ce qui permet d'obtenir sur le corpus de test d'ESTER-2 une réduction du DER d'environ 2,43 points absolus.

Remerciements

Les auteurs remercient Pierre-Michel Bousquet pour l'aide apportée sur l'algorithme de conditionnement des i-vecteurs.

Ces travaux ont été en partie financés par l'Agence Nationale de Recherche (ANR) par l'intermédiaire du projet SODA (ANR-2010-CORD-101-01).

Références

Barras, C., Zhu, X., Meignier, S. et Gauvain, J.-L. (2006). Multi-stage speaker diarization of broadcast news. *IEEE Transactions on Audio, Speech & Language Processing*.

Bousquet, P.-M., Matrouf, D. et Bonastre, J.-F. (2011). Intersession compensation and scoring methods in the i-vectors space for speaker recognition. *In Interspeech*.

Dehak, N., Kenny, P., Dehak, R. et Ouellet, P. (2010). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech & Language Processing*, 19(99):1–23.

Fredouille, C. et Senay, G. (2006). Technical improvements of the E-HMM based speaker diarization system for meeting records. *In* Renals, S., Bengio, S. et Fiscus, J. G., éditeurs : *MLMI*, volume 4299 de *Lecture Notes in Computer Science*, pages 359–370. Springer.

Galliano, S., Gravier, G. et Chaubard, L. (2009). The ESTER 2 evaluation campaign for the rich transcription of French radio broadcasts. *In Interspeech*, pages 2583–2586. ISCA.

Garcia-Romero, D. et Espy-Wilson, C. (2011). Analysis of i-vector length normalization in speaker recognition systems. *In Interspeech*.

Kenny, P., Boulianne, G., Ouellet, P. et Dumouchel, P. (2007). Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech & Language Processing*, 15(4):1435–1447.

Le, V.-B., Mella, O. et Fohr, D. (2007). Speaker diarization using normalized cross likelihood ratio. *In Interspeech*, pages 1869–1872. ISCA.

Meignier, S. et Merlin, T. (2010). LIUM SpkDiarization : An open-source toolkit for diarization. *In CMU SPUD Workshop*.