



# Séparation de sources: quand l'acoustique rencontre le machine learning

Emmanuel Vincent

## ► To cite this version:

Emmanuel Vincent. Séparation de sources: quand l'acoustique rencontre le machine learning. 13e Congrès Français d'Acoustique, Apr 2016, Le Mans, France. hal-01398720

**HAL Id: hal-01398720**

**<https://inria.hal.science/hal-01398720>**

Submitted on 1 Dec 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# SÉPARATION DE SOURCES : QUAND L'ACOUSTIQUE RENCONTRE LE MACHINE LEARNING

Emmanuel Vincent, Inria Nancy – Grand Est

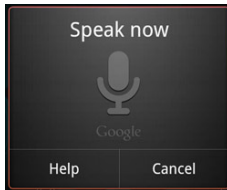
# La séparation de sources: qu'est-ce que c'est?

But: extraire les signaux correspondant aux différentes sources sonores présentes simultanément dans un enregistrement.

À quoi ça sert?

- écouter les sources séparées,
- les remixer,
- en extraire de l'information.

# Communication parlée



La séparation de sources permet

- de sélectionner la source d'intérêt et réduire le bruit,
- d'améliorer la reconnaissance de la parole,

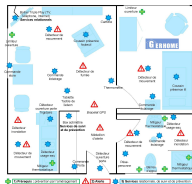
# Ingénierie sonore et écoute interactive



La séparation de sources permet

- d'upmixer des contenus mono/stéréo en format multicanal,
- de remixer ces contenus en studio ou à l'écoute,

# Commande vocale à distance et monitoring sonore



La séparation de sources permet

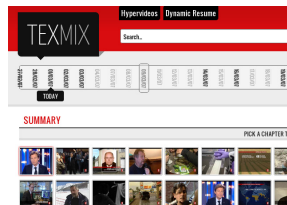
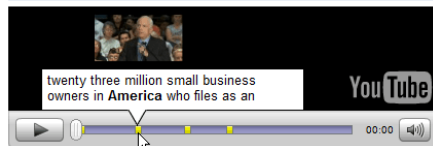
- de commander à distance les appareils de la maison connectée,
- de détecter des sons particuliers dans une scène sonore,

# Contenus audiovisuels

What did the candidates say?

Search Videos

All Politicians | [McCain](#) | [Obama](#)



La séparation de sources permet

- de mieux reconnaître la parole et le locuteur dans les documents parlés, et ainsi de mieux les indexer.

## Et la tomographie et l'holographie dans tout ça?

Comme la tomographie et l'holographie, la séparation de sources est un **problème inverse**.

Mais les dimensions et les informations a priori diffèrent. . .

	Tomographie	Holographie en champ proche	Séparation de sources
Nombre de micros	moyen	élevé	<b>faible</b>
Nombre de sources	1	élevé	<b>moyen</b>
Niveau de bruit	élevé	faible	<b>variable</b>
Infos sur les canaux	inconnus	connus	<b>modèle</b>
Infos sur les sources	connues	inconnues	<b>modèle</b>

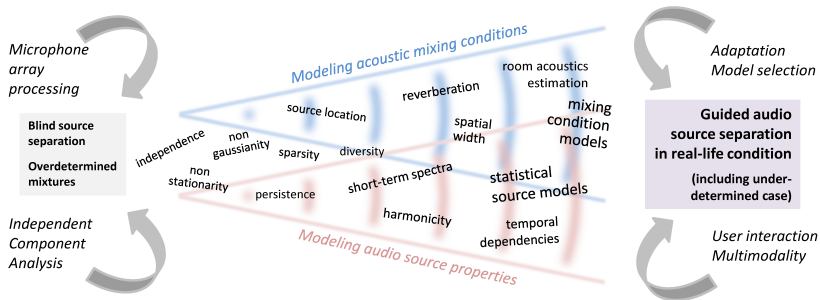


# Niveaux d'information sur les canaux et les sources

On parle de séparation de sources

- **aveugle**: pas d'info a priori (inapplicable à l'audio)
- **faiblement guidée**: info générale liée au contexte d'usage, par exemple "les sources sont de la parole"
- **fortement guidée**: info spécifique au signal traité: position spatiale des sources, identité du locuteur, partition musicale. . .
- **informée**: info très précise encodée et transmise avec l'audio (codage audio multicanal flexible)

# Évolution des recherches



# PRINCIPES GÉNÉRAUX

# De la séparation générale à la séparation audio

Avant 2005, formulation comme un problème inverse linéaire:

$$\mathbf{x}_t = \sum_{\tau=0}^{\infty} \mathbf{A}_{\tau} \mathbf{s}_{t-\tau}$$

$\mathbf{x}_t$ :  $I \times 1$  mélange  
 $\mathbf{s}_t$ :  $J \times 1$  sources (ponctuelles)  
 $\mathbf{A}$ :  $I \times J$  canal  
 $t$ : temps (discret)

Remplacée par la formulation plus générale suivante:

$$\mathbf{x}_{tf} = \sum_{j=1}^J \mathbf{y}_{jtf}$$

$\mathbf{y}_{jtf}$ :  $I \times 1$  **image spatiale** de la source  $j$   
(peut être diffuse)  
 $t$ : temps  
 $f$ : fréquence

But: répartir le signal  $\mathbf{x}_{tf}$  en chaque point temps-fréquence entre les différentes sources.

# Modèle gaussien non-stationnaire

Comment modéliser  $\mathbf{y}_{jtf}$ ?

Théorème: impossible de séparer deux bruits blancs gaussiens stationnaires.

Modèle non-gaussien populaire jusqu'en 2010.

Modèle gaussien non-stationnaire le plus utilisé aujourd'hui:

$$\mathbf{y}_{jtf} \sim \mathcal{N}(\mathbf{0}, v_{jtf} \mathbf{R}_{jf})$$

$\mathcal{N}(\cdot)$ : gaussienne complexe multivariée  
 $v_{jtf}$ : spectre de puissance  
 $\mathbf{R}_{jf}$ : matrice de covariance spatiale

# Matrice de covariance spatiale

La matrice de covariance spatiale encode les trois indices de la perception spatiale (étudiés notamment en psycho-acoustique):

$$\mathbf{R}_{jf} = \begin{pmatrix} r_{11} & r_{12}e^{-i\varphi} \\ r_{12}e^{i\varphi} & r_{22} \end{pmatrix}$$

- la **différence d'intensité** intercanale  $10 \log_{10}(r_{22}/r_{11})$
- la **différence de phase** intercanale  $\varphi$
- la **cohérence** intercanale  $r_{12}/\sqrt{r_{11}r_{22}}$

# Séparation en deux étapes

- Estimation des paramètres (maximum a posteriori):

$$\max_{\theta} \sum_{t,f} \log p(\theta | \mathbf{x}_{tf})$$

où  $\theta = \{\mathbf{R}_{jf}, v_{jtf}\}$ .

- Estimation des sources (erreur quadratique minimale):

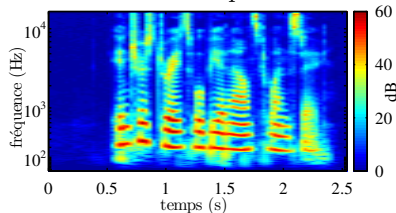
$$\hat{\mathbf{y}}_{jtf} = \boldsymbol{\Omega}_{jtf} \mathbf{x}_{tf} \quad \text{où} \quad \boldsymbol{\Omega}_{jtf} = v_{jtf} \mathbf{R}_{jf} (\sum_{j'} v_{j'tf} \mathbf{R}_{j'f})^{-1}$$

$\boldsymbol{\Omega}_{jtf}$  est appelé **filtre de Wiener**.

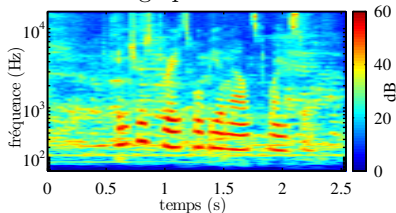
# Filtre de Wiener mono

En mono, le filtre opère comme un masque temps-fréquence.

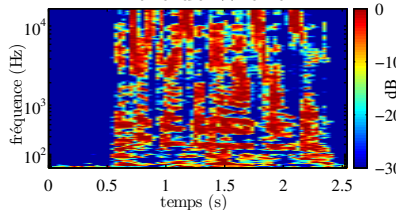
Source de parole



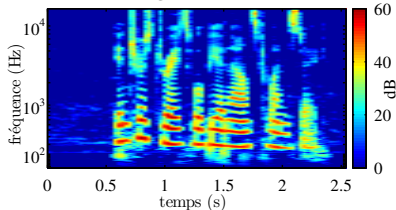
Mélange parole + bruit



Filtre de Wiener



Signal filtré



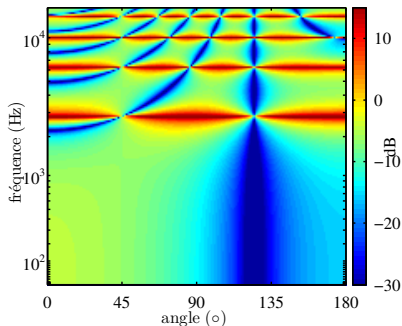


# Filtre de Wiener multicanal

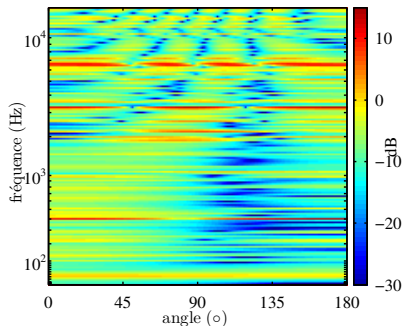
En multicanal, le filtre effectue conjointement:

- un filtrage spectral (masque temps-fréquence)
- un filtrage spatial (formation de voies adaptative).

Filtre de Wiener (anéchoïque)



Filtre de Wiener (réverb)



## Formulation explicite du critère d'estimation

Comme toutes les sources sont gaussiennes, leur somme l'est aussi:

$$\mathbf{x}_{tf} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{x}_{tf}}) \quad \text{avec} \quad \mathbf{\Sigma}_{\mathbf{x}_{tf}} = \sum_j v_{jtf} \mathbf{R}_{jf}$$

Le critère du maximum a posteriori se calcule explicitement:

$$\begin{aligned} \max_{\theta} \sum_{t,f} \log p(\theta | \mathbf{x}_{tf}) &= \max_{\theta} \left( \log p(\theta) + \sum_{t,f} \log p(\mathbf{x}_{tf} | \theta) \right) \\ &= \max_{\theta} \left( \log p(\theta) - \sum_{t,f} \log \det(\mathbf{\Sigma}_{\mathbf{x}_{tf}}) + \text{tr}(\mathbf{\Sigma}_{\mathbf{x}_{tf}}^{-1} \hat{\mathbf{\Sigma}}_{\mathbf{x}_{tf}}) \right) \end{aligned}$$

avec  $\hat{\mathbf{\Sigma}}_{\mathbf{x}_{tf}} = \mathbf{x}_{tf} \mathbf{x}_{tf}^H$  la matrice de covariance du mélange observé.

# Algorithme EM général

Comment estimer à la fois  $v_{jtf}$  et  $\mathbf{R}_{jf}$  (pas de solution analytique)?

Algorithme itératif **espérance-maximisation** (EM):

- étape E: on estime les sources en fonction des paramètres précédents  $\theta^*$

$$\mathbf{y}_{jtf} | \mathbf{x}_{tf}, \theta^* \sim \mathcal{N}(\boldsymbol{\Omega}_{jtf} \mathbf{x}_{tf}, (\mathbf{I} - \boldsymbol{\Omega}_{jtf}) v_{jtf} \mathbf{R}_{jf})$$

- étape M: on met à jour les paramètres en fonction des sources

$$\max_{\theta} \mathbb{E}_{\mathbf{y}_{jtf} | \mathbf{x}_{tf}, \theta^*} \left( \sum_{j,t,f} \log p(\theta | \mathbf{y}_{jtf}) \right)$$

En pratique, converge vers un optimum local.

## Algorithme EM (modèle non contraint)

Dans le cas où  $v_{jtf}$  et  $\mathbf{R}_{jf}$  ne sont pas contraints, on obtient:

■ étape E:

$$\Omega_{jtf} = v_{jtf} \mathbf{R}_{jf} (\sum_{j'} v_{j'tf} \mathbf{R}_{j'f})^{-1}$$

$$\hat{\mathbf{R}}_{y_{jtf}} = \Omega_{jtf} \hat{\mathbf{R}}_{x_{tf}} \Omega_{jtf}^H + (\mathbf{I} - \Omega_{jtf}) v_{jtf} \mathbf{R}_{jf}$$

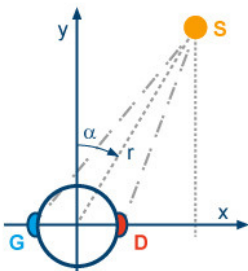
■ étape M:

$$\mathbf{R}_{jf} \leftarrow \frac{1}{T} \sum_t \frac{\hat{\mathbf{R}}_{y_{jtf}}}{v_{jtf}}$$

$$v_{jtf} \leftarrow \text{tr}(\mathbf{R}_{jf}^{-1} \hat{\mathbf{R}}_{y_{jtf}}) / I$$

# MODÉLISATION SPATIALE

## Vecteur de direction (cas anéchoïque)



En champ anéchoïque, on aurait  $\mathbf{R}_{jf} = \mathbf{d}_{jf} \mathbf{d}_{jf}^H$  où  $\mathbf{d}_{jf}$  est le **vecteur de direction**:

$$\mathbf{d}_{jf} = \left[ \frac{1}{r_{1j}} e^{-2i\pi f r_{1j}/c}, \dots, \frac{1}{r_{lj}} e^{-2i\pi f r_{lj}/c} \right]^T$$

$c$ : vitesse du son  
 $r_{ij}$ : distance source  $j$   
au micro  $i$

## Valeur moyenne de la covariance spatiale (cas réverbérant)

Les échos et la réverbération modifient la direction apparente et réduisent la cohérence entre les canaux.

La théorie statistique de l'acoustique des salles montre que  $\mathbf{R}_{jf}$  vaut en moyenne

$$\mu_{\mathbf{R}_{jf}} = \mathbf{d}_{jf} \mathbf{d}_{jf}^H + \sigma_{\text{ech}}^2 \mathbf{\Omega}_f$$

$\mathbf{d}_{jf}$ : vecteur de direction  
 $\sigma_{\text{ech}}^2$ : puissance du champ réfléchi  
 $\mathbf{\Omega}_f$ : covariance spatiale d'un champ isotrope (forme analytique)

Permet de définir une distribution a priori  $p(\theta)$ , peu utilisée en pratique (nécessite la position relative des sources).

# Recherches actuelles

- estimer conjointement la position des sources et les matrices de covariance spatiale associées,
- modéliser l'effet de la réverbération d'une trame temporelle sur les suivantes,
- modéliser la réponse de salle entre les sources et les micros en interpolant les réponses enregistrées en des points voisins,
- mieux modéliser les sources et micros mobiles



# MODÉLISATION SPECTRALE: NMF

# Factorisation matricielle positive

Modèle populaire: factorisation matricielle positive (NMF)

$$v_{jtf} = \sum_{k=1}^K w_{jkf} h_{jkt}$$

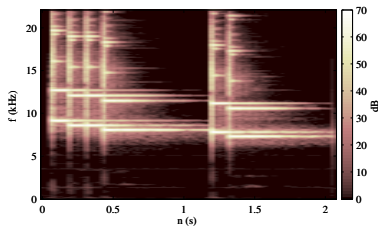
$w_{jkf}$ : spectre de base  
 $h_{jkt}$ : coefficient d'activation

Les spectres de base  $w_{jkf}$  peuvent être appris

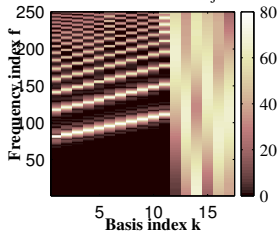
- soit préalablement sur un corpus de sources isolées,
- soit à partir du mélange à séparer.

# Exemple

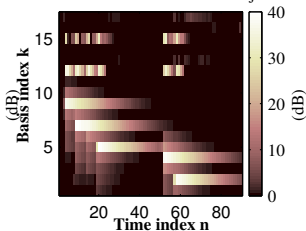
Source musicale (xylophone)



Basis spectra  $W_j$



Temporal activations  $H_j$



# Algorithme NMF multicanal

- étape E (inchangée):

$$\Omega_{jtf} = v_{jtf} \mathbf{R}_{jf} (\sum_{j'} v_{j'tf} \mathbf{R}_{j'f})^{-1}$$

$$\hat{\mathbf{R}}_{y_{jtf}} = \Omega_{jtf} \hat{\mathbf{R}}_{x_{tf}} \Omega_{jtf}^H + (\mathbf{I} - \Omega_{jtf}) v_{jtf} \mathbf{R}_{jf}$$

- étape M:

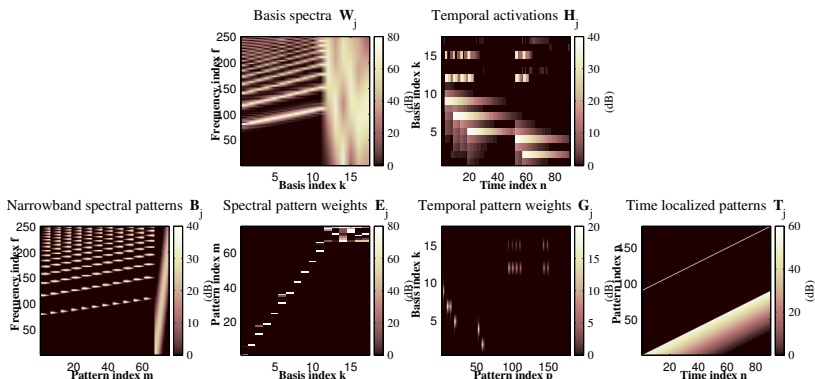
$$\mathbf{R}_{jf} \leftarrow \frac{1}{T} \sum_t \frac{\hat{\mathbf{R}}_{y_{jtf}}}{v_{jtf}}$$

$$\xi_{jtf} \leftarrow \text{tr}(\mathbf{R}_{jf}^{-1} \hat{\mathbf{R}}_{y_{jtf}}) / I \quad (\text{spectre non contraint, inchangé})$$

$$h_{kt} \leftarrow h_{kt} \frac{\sum_f w_{kf} v_{jtf}^{-2} \xi_{jtf}}{\sum_f w_{kf} v_{jtf}^{-1}} \quad (\text{NMF, mise à jour « multiplicative »})$$

# Pour aller plus loin: modèles spectraux avancés

- modèle source-filtre
- décomposition des spectres de base et des activations en coefficients de structure fine et d'enveloppe.



## Pour aller plus loin: modèles temporels avancés

- a priori de continuité/parcimonie sur  $h_{jkt}$
- spectrogrammes de base (au lieu de simples spectres)
- modèles de Markov sur  $h_{jkt}$

# Recherches actuelles

- modéliser le spectre de phase,
- exploiter les redondances (jingles ou musiques de fond, contenus multilingues. . . ),
- lorsque c'est possible, interagir avec l'ingénieur du son pour adapter/améliorer le modèle.

# MODÉLISATION SPECTRALE: DNN



# Réseaux de neurones profonds (DNN)

Révolution en apprentissage automatique depuis 2006...

...et en audio depuis 2010!

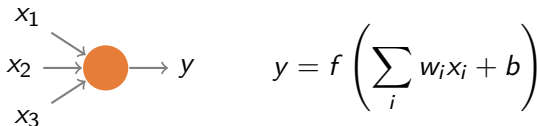
Un DNN est une fonction non-linéaire multivariée.

Représente le traitement complet (modélisation + estimation des paramètres + filtrage): plus besoin de modèle!

# Neurone

Neurone: fonction non-linéaire paramétrique simple.

Par ex.: transformation linéaire + fonction non-linéaire scalaire.

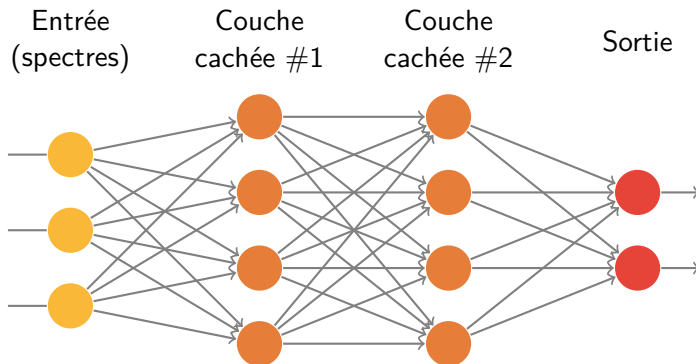


- sigmoïde  $f(x) = 1/(1 + e^{-x})$
- rectificatrice  $f(x) = \max(x, 0)$

Certains neurones représentent des fonctions plus compliquées (LSTM, GRU) ou ont plusieurs sorties (*softmax*).

# Réseau de neurones

Perceptron multicouches (profond si  $\geq 3$  couches cachées):



Il existe aussi des DNN récurrents qui exploitent la valeur passée de chaque neurone.

# Apprentissage

Paramètres: poids  $w_i$  et biais  $b$  de tous les neurones.

Données: séquence d'entrées  $\mathbf{x}_t$  et de sorties désirées  $\mathbf{y}_t$ .

Apprentissage: minimiser une fonction de coût  $c(\hat{\mathbf{y}}_t, \mathbf{y}_t)$  par descente de gradient

- initialisation aléatoire des paramètres,
- calcul récursif du gradient par la formule de *rétropropagation*,
- somme sur un *minibatch* et mise à jour des paramètres,
- plusieurs passes sur les données (*époques*),
- arrêt quand le coût ne décroît plus sur des données disjointes.

Lourd, requiert une implémentation sur carte graphique (GPU).

# Test

Données: séquence d'entrées  $\mathbf{x}_t$ .

Test: calcul des sorties  $\hat{\mathbf{y}}_t$  (*forward pass*).

Peut tourner en temps réel.

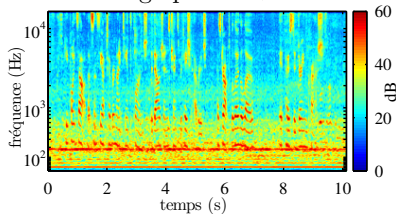
# Avantages théoriques

Par rapport aux algorithmes précédents basées sur des modèles:

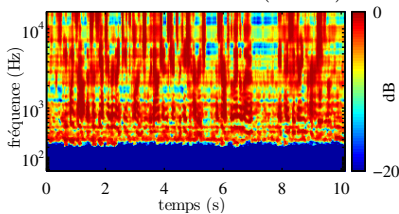
- peut modéliser des caractéristiques plus complexes,
- tire mieux parti des grandes quantités de données disponibles,
- plus invariant aux valeurs aberrantes observées,
- facile à entraîner de façon discriminante, c'est-à-dire pour maximiser directement la performance de la tâche souhaitée.

# Exemple

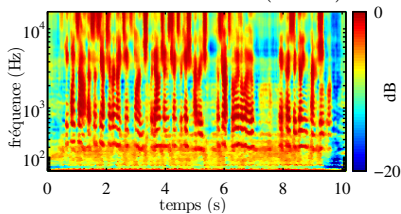
Mélange parole + bruit



Filtre de Wiener (NMF)



Filtre de Wiener (DNN)



# Algorithme DNN multicanal

- étape E (inchangée):

$$\Omega_{jtf} = v_{jtf} \mathbf{R}_{jf} (\sum_{j'} v_{j'tf} \mathbf{R}_{j'f})^{-1}$$

$$\hat{\mathbf{R}}_{y_{jtf}} = \Omega_{jtf} \hat{\mathbf{R}}_{x_{tf}} \Omega_{jtf}^H + (\mathbf{I} - \Omega_{jtf}) v_{jtf} \mathbf{R}_{jf}$$

- étape M:

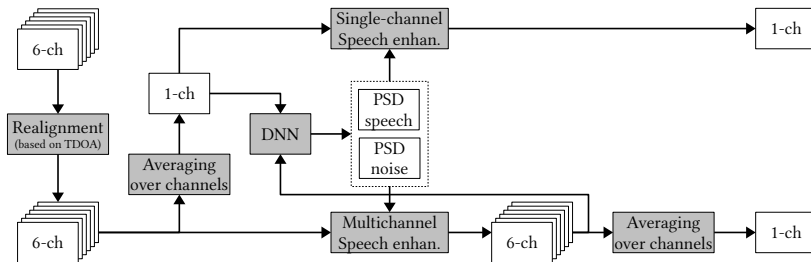
$$\mathbf{R}_{jf} \leftarrow \frac{1}{T} \sum_t \frac{\hat{\mathbf{R}}_{y_{jtf}}}{v_{jtf}}$$

$$\xi_{jtf} \leftarrow \text{tr}(\mathbf{R}_{jf}^{-1} \hat{\mathbf{R}}_{y_{jtf}}) / I \quad (\text{spectre non contraint, inchangé})$$

$$v_{jtf} \leftarrow \text{DNN}(\xi_{jtf}^{1/2})^2 \quad (\text{réestimation par DNN})$$



# Schéma de traitement





# Résultats (parole)

Noisy	 	WER=33.23%
Single-channel DNN	 	WER=36.92%
Delay-and-sum	 	WER=26.30%
DNN post-filter	 	WER=26.54%
Multichannel DNN	 	WER=20.17%

CHiME-3: parole enregistrée dans un bus. Une seule itération de DNN, pas de post-traitement. Reconnaissance de la parole par GMM-HMM multi-conditions.

## Résultats (musique)

Angela Thomas Wade - *Milk Cow Blues* 

Voix chantée estimée 

# Recherches actuelles

- améliorer la qualité pour la tâche visée par post-traitement,
- adapter le DNN aux signaux de test,
- mieux simuler les données nécessaires à l'apprentissage,
- introduire les connaissances issues des modèles précédents.

# CONCLUSION

# Résumé

La séparation de sources est un problème inverse.

Pour le résoudre, on emprunte des éléments

- à l'acoustique: acoustique des salles, psycho-acoustique, production de la parole. . .
- à l'apprentissage automatique: EM, NMF, DNN. . .

Les DNN amènent un changement radical de paradigme: plus besoin de modèle, on apprend le résultat directement!

Il est probable que ce changement de paradigme émerge bientôt pour d'autres problèmes de l'acoustique. . .

# Références

## Articles liés à ce tutoriel:

- E. Vincent, N. Bertin, R. Gribonval, and F. Bimbot, “From blind to guided audio source separation: How models and side information can improve the separation of sound”, *IEEE SPM*, 31(3), 2014.
- S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, “Multi-microphone speech enhancement and source separation”, overview paper to appear in *IEEE/ACM TASL*, 2016.
- A. A. Nugraha, A. Liutkus, and E. Vincent, “Multichannel audio source separation with deep neural networks”, RR-8740, Inria, 2016.

## Listes de diffusion, corpus, logiciels, campagnes d'évaluation:

- <https://groups.google.com/forum/#!forum/machinelisting>
- <https://wiki.inria.fr/rosp/>
- <https://sisec.inria.fr/>