TRAITEMENT DE LA PAROLE

# RECONNAISSANCE DU LOCUTEUR

Le Mans Université

Anthony Larcher - Le Mans Université - 2019

# PLAN DU COURS

Les mixtures de Gaussiennes (GMM)

- Motivations
- Le modèle

Utilisation des GMM pour la reconnaissance du locuteur

- Le modèle du monde (UBM)
- L'adaptation au locuteur (MAP)
- L'hypothèse alternative

Et quoi d'autre?

# CONTEXTE HISTORIQUE: DES MODÈLES À MIXTURES DE GAUSSIENNES

- 1995 - 2016: modèles génératifs à base de Gaussiennes
- Grandes avancées en termes de performances et robustesse
- Premières applications commerciales (mobile, banques, contrôle d'accès)
- Utilisations industrielles (call-centre, agent de conversation)
- Aujourd'hui? on fait mieux mais c'est encore utilisé

# RAPPEL: CLASSIFICATION PAR APPROCHES GÉNÉRATIVES

Exemple de la modélisation Gaussienne [1]

- On fait l'hypothèse que les échantillons observés donnent des renseignements sur leur voisinage
- À partir de quelques échantillons, on peut estimer la probabilité d'une nouvelle observation d'appartenir à la classe cible

[1] F. Bimbot, I. Magrin-Chagnolleau et L. Mathan, Second-order statistical measures for text-independent speaker identification, in Speech Communication, 1995, vol. 17, no 1-2, p 177-192

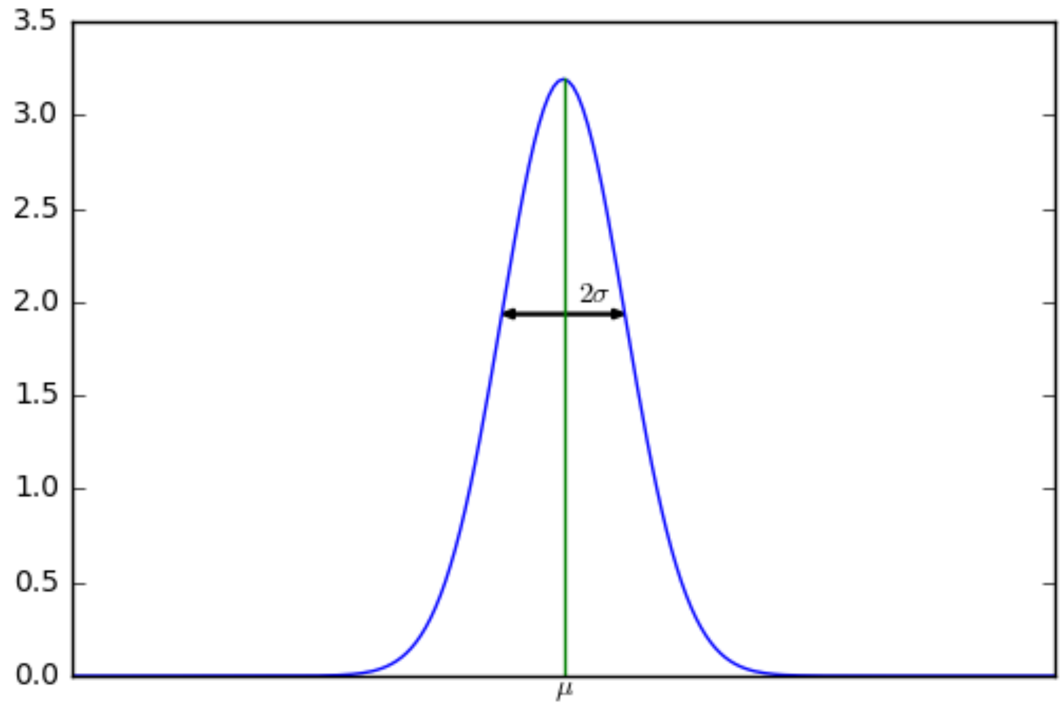
# CLASSIFICATION PAR APPROCHES GÉNÉRATIVES

Pourquoi la modélisation Gaussienne?

- mathématiquement « facile »
- paramètres faciles à estimer:
  - moyenne
  - variance
- Théorème Central-limite
  les ***MOYENNES*** d'échantillons indépendants qui suivent une même loi de probabilité tendent vers une distribution normale pour peu qu'elles soient suffisamment nombreuses.

# CLASSIFICATION PAR APPROCHES GÉNÉRATIVES

## Exemple de la modélisation Gaussienne



# MODÈLES GAUSSIENS ET COMPLEXITÉ

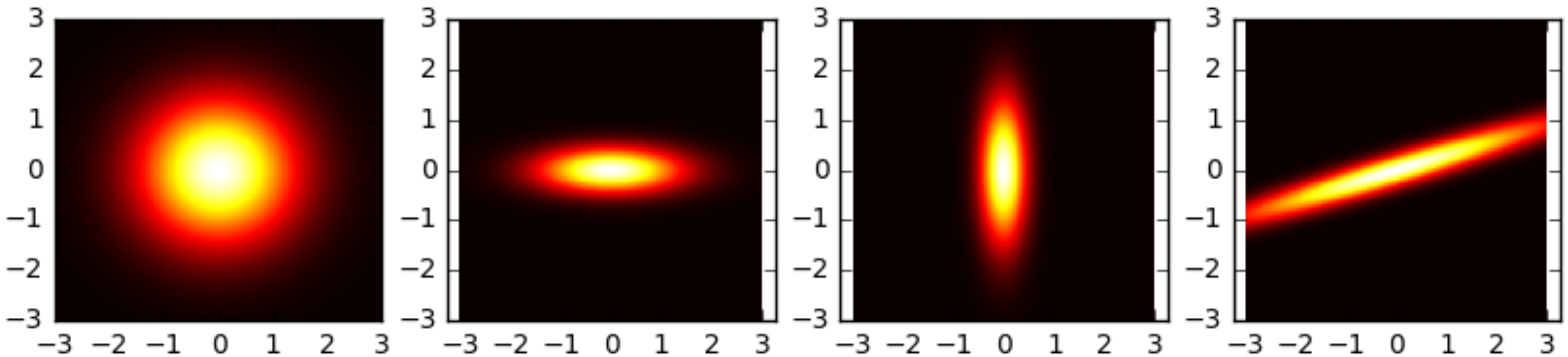
$$P(o|X) = \frac{1}{\sqrt{2\Pi}|\Sigma|} \exp\left(-\frac{1}{2}(o-\mu)\Sigma^{-1}(o-\mu)^T\right)$$

Une Gaussienne:

- vecteur de moyenne: dimension de l'espace: N
- matrice de covariance: dimension de l'espace au carré: $N^2$
- au total: N + N( N + 1)/2 paramètres à estimer pour modéliser un locuteur
  (la matrice de covariance est symétrique, définie positive)

- Différentes options possibles pour la matrice de covariance:
  - pleine: N(N + 1)/2 paramètres
  - diagonale: N paramètre
  - sphérique: 1 paramètres

# MODÈLES GAUSSIENS ET COMPLEXITÉ




- **<u>Covariance sphérique</u>**: multiple de la matrice identité
  variables indépendantes et les surfaces iso-probables sont des hyper-sphères

- **<u>Covariance diagonale</u>**: zéros partout sauf sur au moins quelques termes de la diagonale. Variables indépendantes mais avec différentes échelles. Les surfaces équiprobables sont des hyper-ellipsoïdes dont les axes principaux sont alignés sur les axes du repère utilisé.

- **<u>Covariance pleine:</u>** Les variables sont dépendantes, Les surfaces équiprobables sont des hyper-ellipsoïdes sans alignement spécifique

# MODÈLES GAUSSIENS ET COMPLEXITÉ

## Estimation des paramètres d'une Gaussienne

- Soit une séquence $O = \{o_t\}$ de vecteurs acoustiques observés pour un même locuteur.
- Les paramètre de la Gaussienne sont estimés directement comme suit:

$$\mu = \frac{1}{T+1}\sum_{t=0}^{T} o_t \qquad \Sigma = \frac{1}{T+1}\sum_{t=0}^{T}(o_t o_t^T)$$

# MODÈLES GAUSSIENS ET COMPLEXITÉ

Limitations des Gaussiennes multi-variées:

- distribution uni-modale
- pas robustes aux cas particuliers
- beaucoup de paramètres à estimer si on travaille dans un espace de grande dimension

# MODÈLES À MIXTURES DE GAUSSIENNES

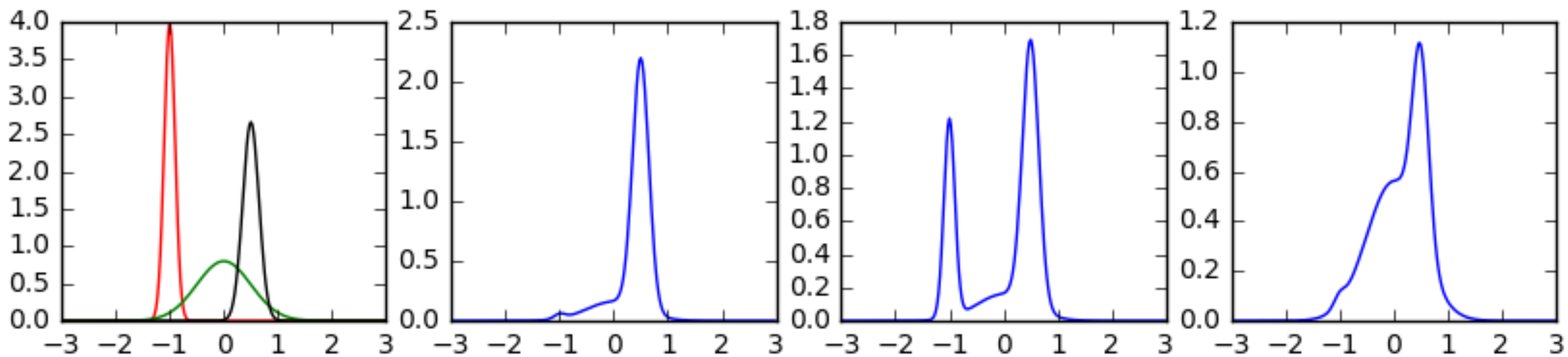
Limitations des Gaussiennes multi-variées:

- distribution uni-modale: les mixtures de Gaussiennes peuvent modéliser des distributions multi-modales
- pas robustes aux cas particuliers: les distributions Student-t sont robustes aux cas particuliers
- beaucoup de paramètres à estimer si on travaille dans un espace de grande dimension: les modèles de sous-espace (PPCA, Factor Analysis) réduisent le nombre de paramètres à estimer

# MODÈLES À MIXTURES DE GAUSSIENNES

- En locuteur depuis 1995 on a choisit les GMMs:

$$P(o|\Lambda) = \sum_{c}^{C} w_c \mathcal{N}(\mu_c, \Sigma_c) \quad \text{avec} \quad \sum_{c}^{C} w_c = 1 \ \text{ et } \ w_c \geq 0$$

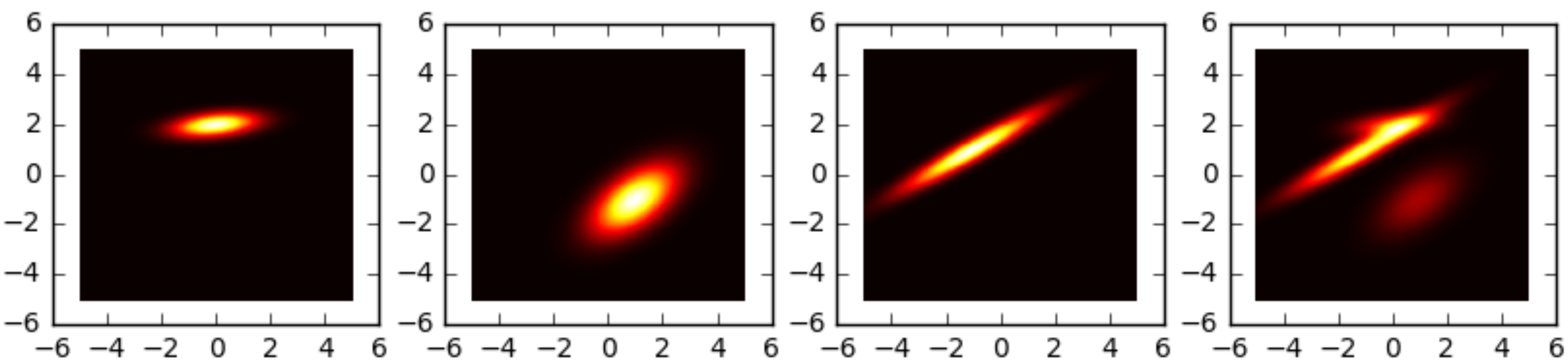



GMMs obtenus pour 3 Gaussiennes et différents vecteurs de poids.

# MODÈLES À MIXTURES DE GAUSSIENNES

- Exemple de GMM à 3 Gaussiennes en 2-dimensions

$$P(o|\Lambda) = \sum_{c}^{C} w_c \mathcal{N}(\mu_c, \Sigma_c) \quad \text{avec} \quad \sum_{c}^{C} w_c = 1 \quad \text{et} \quad w_c \geq 0$$



# MODÈLES À MIXTURES DE GAUSSIENNES

- On peut modéliser des distributions plus complexes

Comment estimer les paramètres du modèle GMM d'un locuteur?

Soit $\theta = \{\mu_c, \Sigma_c, w_c\}_{c=1...C}$ les paramètres du GMM à estimer.

# MODÈLES À MIXTURES DE GAUSSIENNES

Pour estimer les paramètres du modèles: $\hat{\theta}$ , on utilise le
**critère du maximum de vraisemblance.**

On cherche les paramètres $\hat{\theta}$ tels que les données d'apprentissage $\{x_i\}_{i=1}^{I}$ soient les plus vraisemblables.

Pour une observation la vraisemblance est: $Pr(x_i|\theta)$

Pour l'ensemble des données on veut obtenir $\hat{\theta}$ tel que:

$$\hat{\theta} = \underset{\theta}{argmax}[Pr(x_{i \ldots I}|\theta)]$$

hypothèse d'indépendance

$$\hat{\theta} = \underset{\theta}{argmax}[\prod_{i=1}^{I} Pr(x_i|\theta)]$$

# MODÈLES À MIXTURES DE GAUSSIENNES

- Critère du maximum de vraisemblance: on souhaite maximiser une fonction de plusieurs paramètres.
- Approche directe: on dérive par rapport à chaque paramètre, on annule la dérivée et on résout pour trouver ce paramètre.
- Pour une Gaussienne, on utilise le logarithme de la vraisemblance (fonction monotone) qui simplifie les calculs et on obtient les résultats déjà vus.

$$\mu = \frac{1}{T+1}\sum_{t=0}^{T} o_t \qquad \Sigma = \frac{1}{T+1}\sum_{t=0}^{T}(o_t o_t^T)$$

# MODÈLES À MIXTURES DE GAUSSIENNES

- Pour des distributions plus complexes: GMMs, la force brute ne permet pas de trouver les paramètres…
- 2 options:
  1. utiliser un optimiseur
  2. on introduit un variable ***cachée*** ou ***latente***

# MODÈLES À MIXTURES DE GAUSSIENNES

Utilisation des variables latentes:

- On exprime la densité de probabilité $Pr(x)$ comme la marginalisation d'une densité de probabilité conjointe entre x et h, $Pr(x,h)$ ,de sorte que:

$$Pr(x|\theta) = \int Pr(x,h|\theta)dh$$

- On utilise le théorème d'Expectation Maximization (EM) pour trouver les paramètres $\theta = \{\mu_c, \Sigma_c, w_c\}_{c=1\ldots C}$

# MODÈLES À MIXTURES DE GAUSSIENNES

Principe de l'algorithme EM

Permet de trouver les paramètres d'un modèle $\theta$, tel que:

$$\hat{\theta} = \underset{\theta}{argmax} \left[ \sum_{i=1}^{I} \log\left( \int Pr(x_i, h_i|\theta) dh_i \right) \right]$$

1. Initialisation aléatoire des paramètres du modèle
2. Trouver une borne inférieur de la log-vraisemblance (équation ci-dessus)
   La borne inférieure est une fonction paramétrique par $\theta$ et qui est toujours inférieure ou égale à la log-vraisemblance
3. On alterne les étapes E et M jusqu'à convergence

La fonction logarithme est monotone et ne modifie donc pas la position du maximum.
Elle simplifie cependant les calculs.

# MODÈLES À MIXTURES DE GAUSSIENNES

Principe de l'algorithme EM

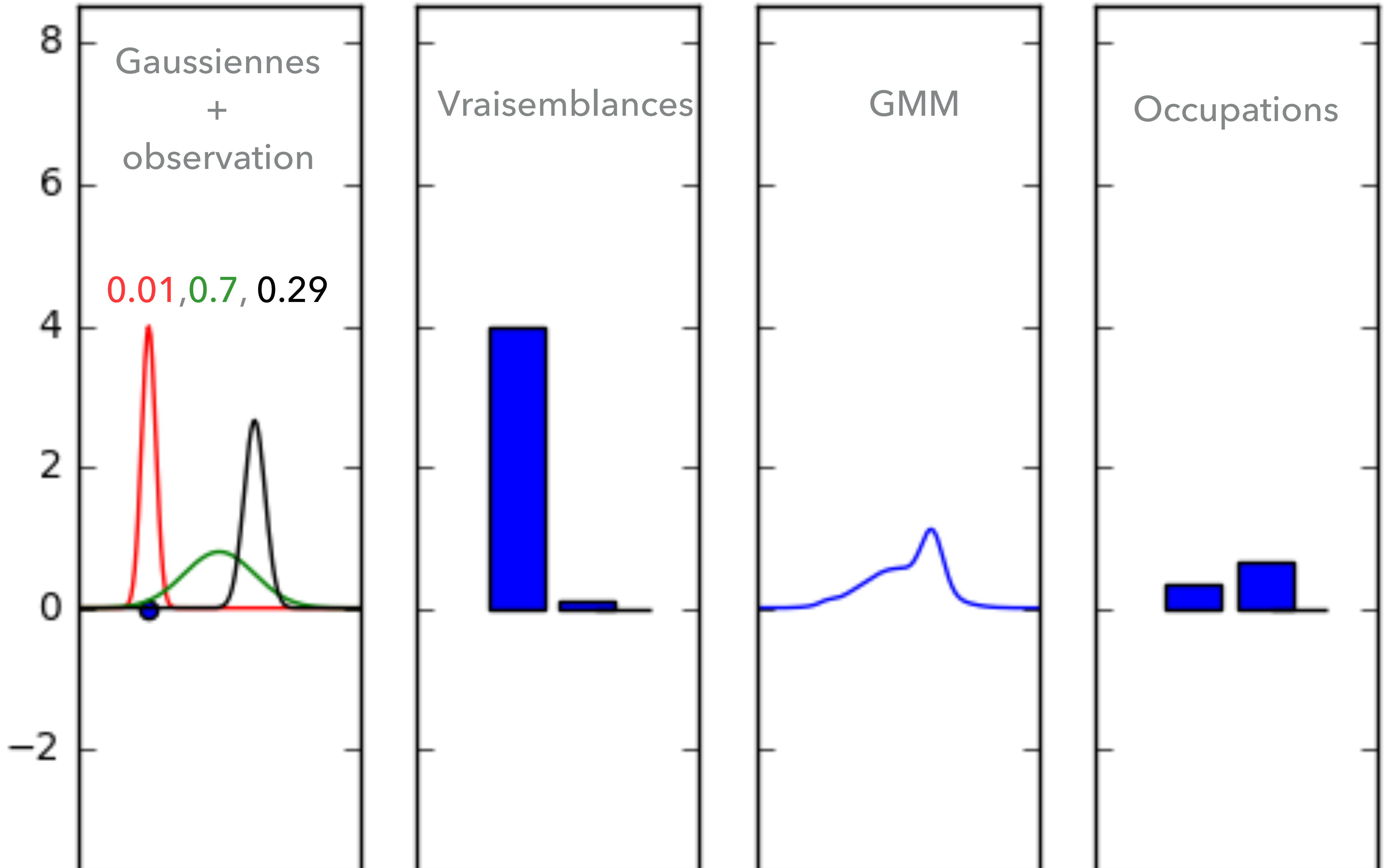
**Expectation**:
on calcule: $\gamma_{ic} = Pr(h_i = c | x_i, \theta^{[t]}) = \dfrac{w_c \mathcal{N}_{x_i}(\mu_c, \Sigma_c)}{\sum_{j=1}^{C} w_j \mathcal{N}_{x_i}(\mu_j, \Sigma_c)}$

La probabilité pour chaque observation d'avoir été générée par la Gaussienne « c » du modèle.

Cette quantité est appelée « occupation » (occupancy)

# MODÈLES À MIXTURES DE GAUSSIENNES

## Occupations (occupancies)



# MODÈLES À MIXTURES DE GAUSSIENNES

Principe de l'algorithme EM

**Maximization**:
on mets à jour les paramètres du modèle.

Les formules de mise à jour sont obtenues en dérivant les expressions, annulant les dérivées par rapport à chaque paramètre et en résolvant.

$$w_c^{t+1} = \frac{\sum_{i=1}^{I} \gamma_{ic}}{\sum_{c=1}^{C} \sum_{i=1}^{I} \gamma_{ic}} \qquad \mu_c^{t+1} = \frac{\sum_{i=1}^{I} \gamma_{ic} x_i}{\sum_{i=1}^{I} \gamma_{ic}}$$

$$\Sigma_c^{t+1} = \frac{\sum_{i=1}^{I} \gamma_{ic}(x_i - \mu_c^{t+1})(x_i - \mu_c^{t+1})^T}{\sum_{i=1}^{I} \gamma_{ic}}$$

# MODÈLES À MIXTURES DE GAUSSIENNES

Analyse de l'EM pour les GMMs:

**Étape M:**
pour chaque observation, on estime son appartenance à une Gaussienne (la vraisemblance que cette observation ait été générée par la dite Gaussienne divisée par la somme des vraisemblance sur toutes les distributions).

$\sum_{i=1}^{I} \gamma_{ic}$ peut être vu comme le nombre d'observation généré par la Gaussienne « c ». Notez que ce nombre peut être un réel (non-entier)

Le poids de la Gaussienne dans le nouveau modèle est directement dépendant du nombre d'observation que cette Gaussienne a « générées ».

$$w_c^{t+1} = \frac{\sum_{i=1}^{I} \gamma_{ic}}{\sum_{c=1}^{C} \sum_{i=1}^{I} \gamma_{ic}}$$

# MODÈLES À MIXTURES DE GAUSSIENNES

## Analyse de l'EM pour les GMMs:

**Étape M:**

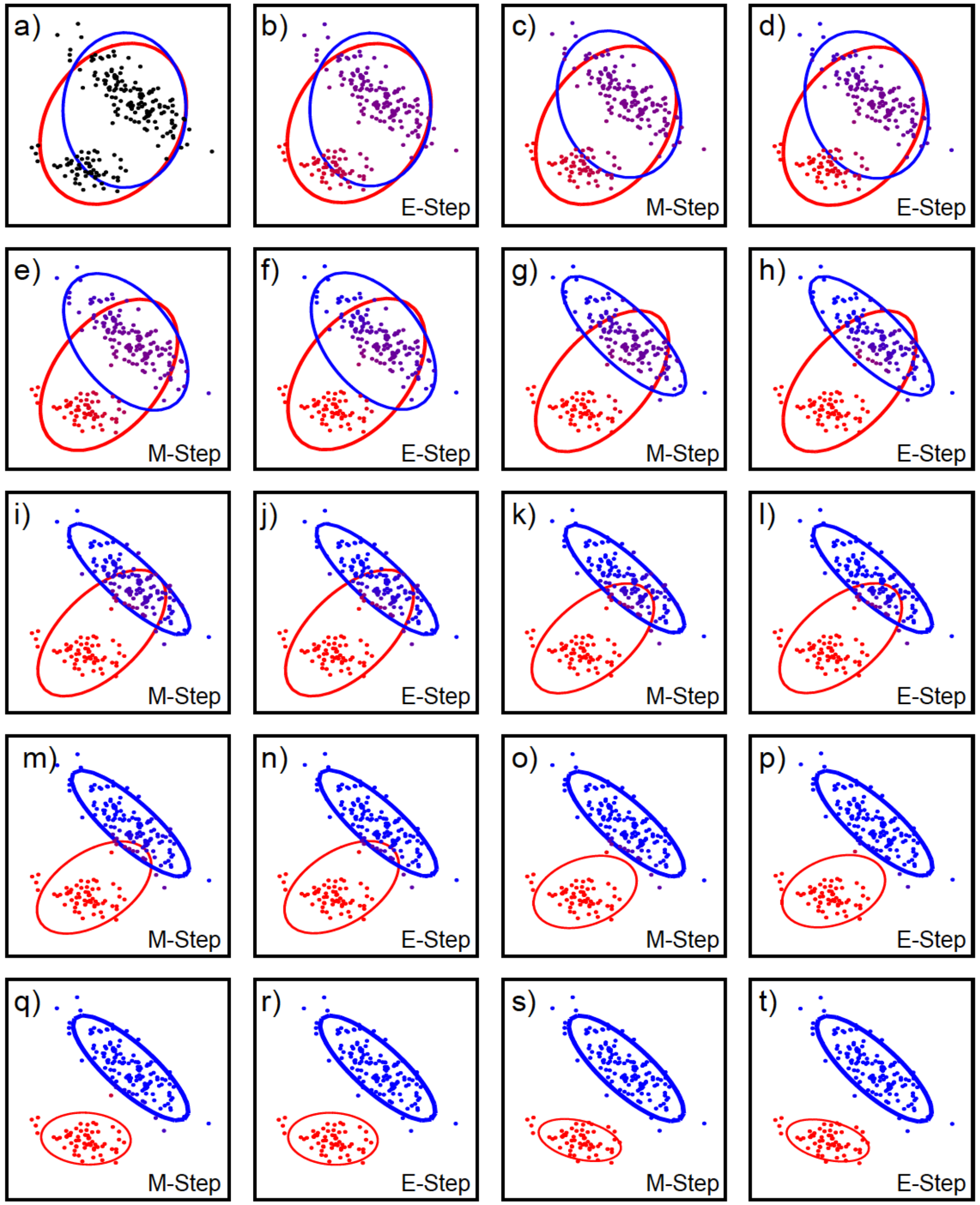
Les nouvelles valeurs de la moyenne et variance de chaque Gaussienne sont similaires au cas d'une Gaussienne seule mais l'influence de chaque observation dans la moyenne et variance est pondérée par son « appartenance » à cette Gaussienne.

$$\mu_c^{t+1} = \frac{\sum_{i=1}^{I} \gamma_{ic} x_i}{\sum_{i=1}^{I} \gamma_{ic}}$$

$$\Sigma_c^{t+1} = \frac{\sum_{i=1}^{I} \gamma_{ic} (x_i - \mu_c^{t+1})(x_i - \mu_c^{t+1})^T}{\sum_{i=1}^{I} \gamma_{ic}}$$

# MODÈLES À MIXTURES DE GAUSSIENNES

Exemple d'apprentissage par EM pour un GMM à 2 Gaussiennes




PRINCE, Simon JD. *Computer vision: models, learning, and inference*. Cambridge University Press, 2012.

# MODÈLES À MIXTURES DE GAUSSIENNES

Rappel sur les occupations

$$\gamma_{ic} = Pr(h_i = c|x_i, \theta^{[t]}) = \frac{w_c \mathcal{N}_{x_i}(\mu_c, \Sigma_c)}{\sum_{j=1}^{C} w_j \mathcal{N}_{x_i}(\mu_j, \Sigma_c)}$$

# MODÈLES À MIXTURES DE GAUSSIENNES

Les statistiques suffisantes: ***sufficient statistics***

En général lorsqu'on travaille avec des modèles Gaussiens, on utilise 3 quantités de façon récurrente: ***les statistiques d'ordre 0, 1 et 2***

| | |
|---|---|
| Statistique d'ordre 0 | $n_c = \sum_{i=1}^{I} \gamma_{ic}$ |
| Statistique d'ordre 1 | $F_c = \frac{1}{n_c} \sum_{i=1}^{I} \gamma_{ic} x_i$ |
| Statistique d'ordre 2 | $S_c = \frac{1}{n_c} \sum_{i=1}^{I} \gamma_{ic} x_i x_i^T$ |

# MODÈLES À MIXTURES DE GAUSSIENNES

## Principe de l'algorithme EM

**Maximization**:
formulation utilisant les *sufficient statistics:*

$$w_c^{t+1} = \frac{n_c}{\sum_{j=1}^{C} n_j}$$

$$\mu_c^{t+1} = \frac{F_c}{n_c}$$

$$\Sigma_c^{t+1} = \frac{S_c}{n_c} - \mu_c^{t+1} {\mu_c^{t+1}}^T$$

# MODÈLES À MIXTURES DE GAUSSIENNES

Interprétation de ce modèle:

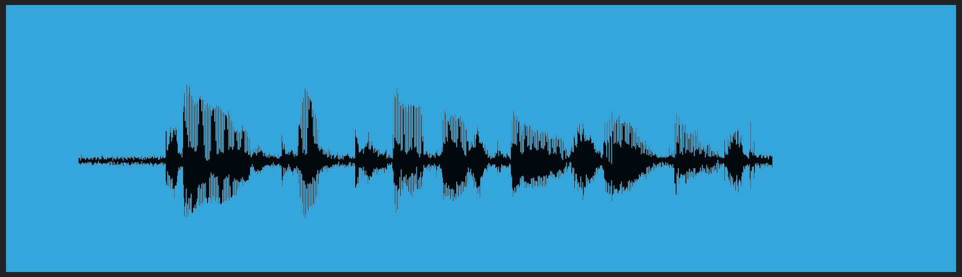
Pour générer des données avec un GMM:

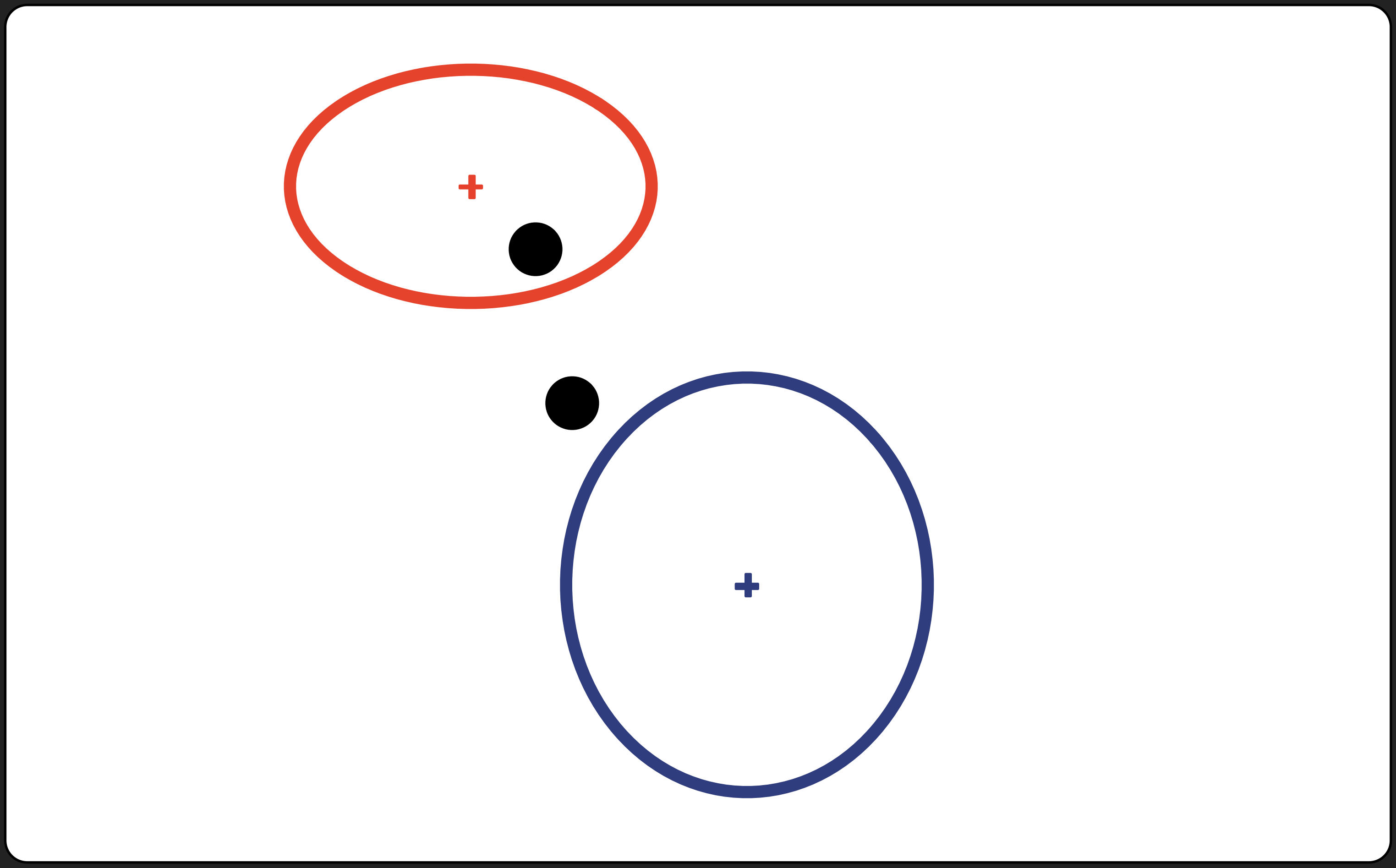
1. On tire aléatoirement la variable latente *h* qui suit une distribution de Bernoulli généralisée.
   Cette valeur nous indique quelle Gaussienne va générer l'observation

2. On tire aléatoirement un variable *x* à partir de la distribution choisie.

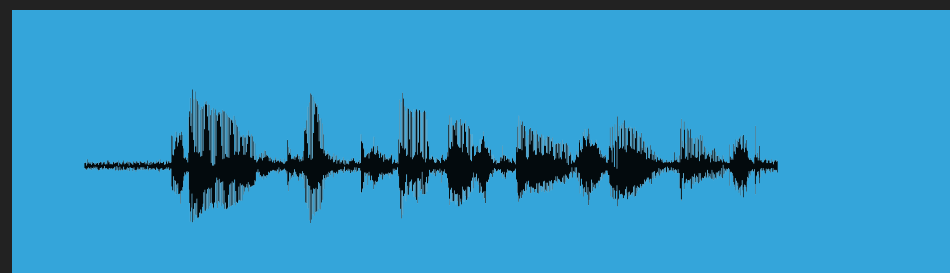
La variable latente a une interprétation simple dans le cas des GMMs.

# MODÈLES À MIXTURES DE GAUSSIENNES

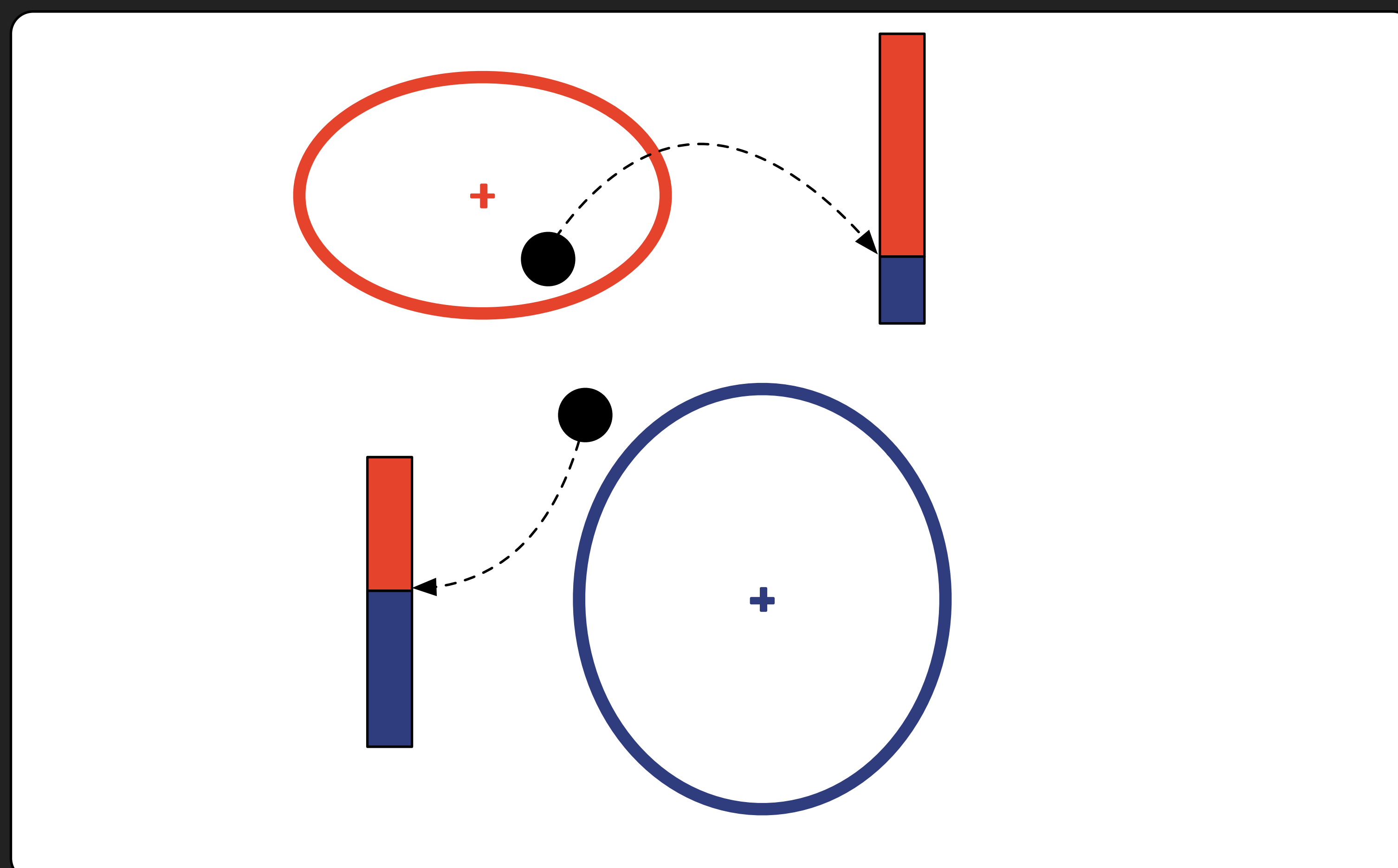




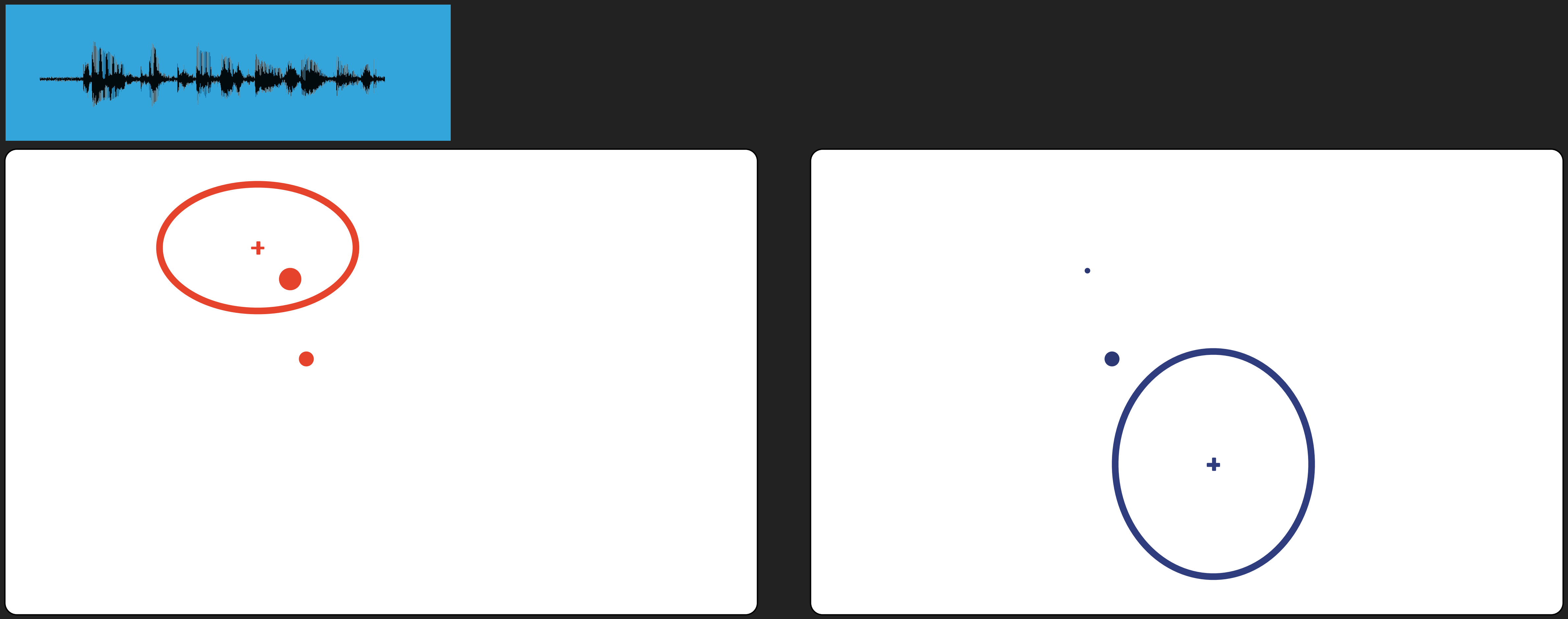
# MODÈLES À MIXTURES DE GAUSSIENNES



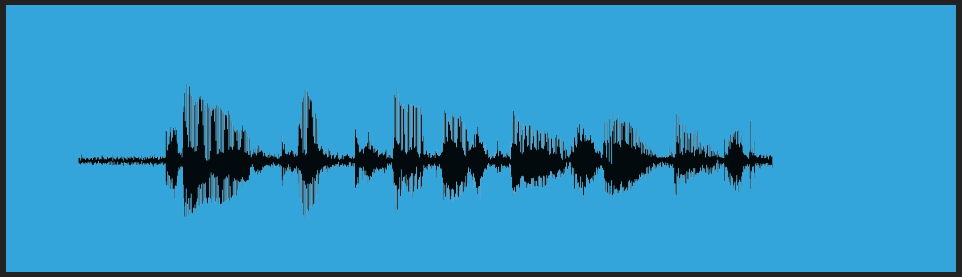
**Statistiques d’ordre 0**

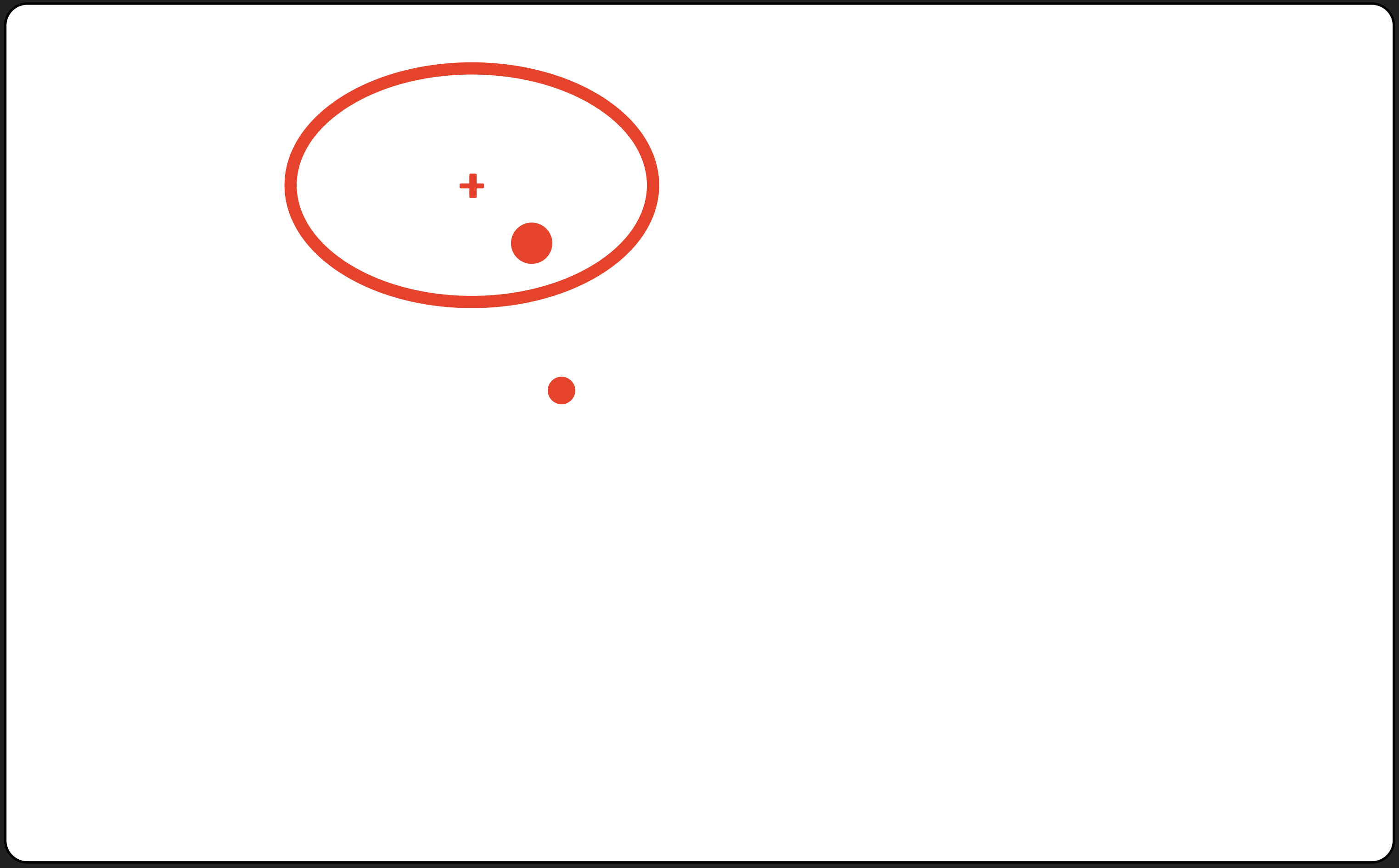


# MODÈLES À MIXTURES DE GAUSSIENNES

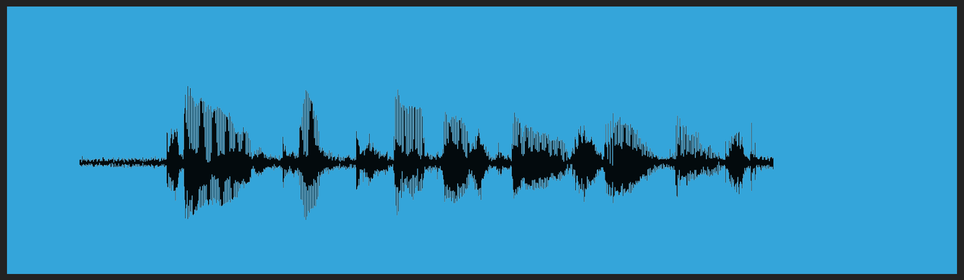


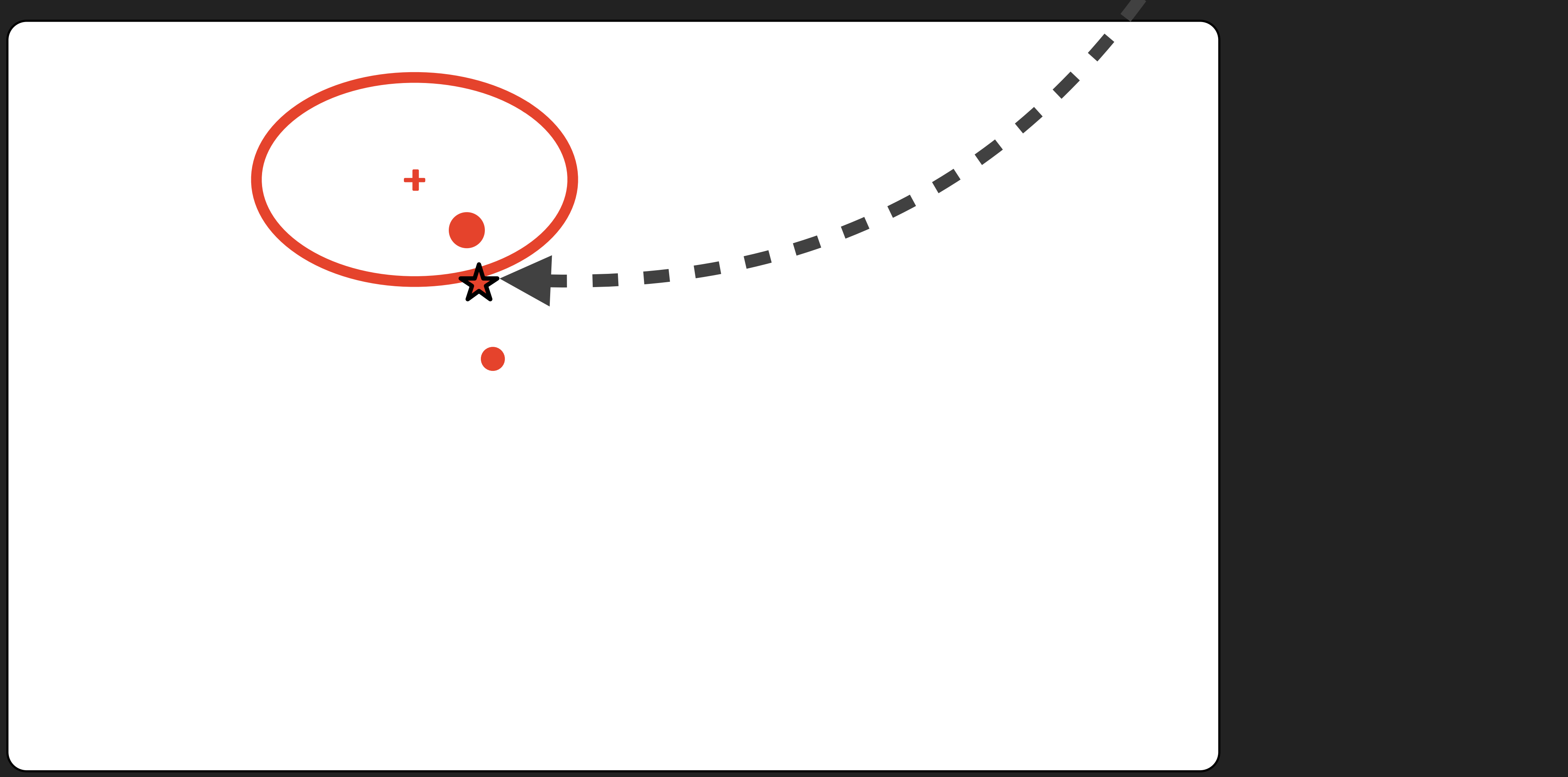
# MODÈLES À MIXTURES DE GAUSSIENNES





# MODÈLES À MIXTURES DE GAUSSIENNES





TRAITEMENT DE LA PAROLE

# UTILISATION DES GMMS EN RECONNAISSANCE DU LOCUTEUR

Le Mans Université

# UTILISATION DES GMM'S EN RECONNAISSANCE DU LOCUTEUR

Objectif premier: modéliser des distributions plus complexes afin d'améliorer la qualité des modèles

1 locuteur = 1 GMM

# UTILISATION DES GMM'S EN RECONNAISSANCE DU LOCUTEUR

- Une Gaussienne: N + N(N+1)/2 paramètres
- GMM à 128 distributions: 128 x (N + N(N + 1)/2 + 1) paramètres
  (avec N =50; 169 728 paramètres)
- Pas assez de données pour apprendre un modèle GMM de façon robuste
- On utilise la plupart du temps des matrices à covariance diagonale pour limiter le nombre de paramètres estimer
  (configuration standard, 2048 distributions, N=50; 2 715 648 paramètres)

<u>Idée</u>:
apprendre un modèle générique qui représente le locuteur moyen puis adapter les paramètres de ce modèle pour apprendre les spécificités de chaque locuteur.

# MODÈLE DU MONDE (UNIVERSAL BACKGROUND MODEL – UBM)

- On apprend un modèle GMM pour modéliser la « voix humaine »
- Intégrer autant de locuteurs que possibles (centaines, milliers)
- Apprentissage avec l'algorithme EM
- Dimension: entre 64 et 8192 distributions
(souvent des puissances de 2 car on peut apprendre en divisant les Gaussiennes)
- Quantité de données nécessaire? > 10h
    - variable selon l'application
    - dépend aussi de la taille du modèle et de la variabilité des données

# ADAPTATION DU MODÈLE DU MONDE AU LOCUTEUR

- Données d'un seul locuteur
- Utilisation d'une information sur la voix humaine a priori
- Si on connait les conditions d'utilisation (téléphone, microphone…) les données d'apprentissage doivent être le plus proche possible pour garantir des performances optimales (mais moins généralisables)
- On souhaite modifier le modèle pour modéliser les informations spécifiques à un locuteur donné
- Approche la plus rependu: Maximum a Posteriori (MAP)

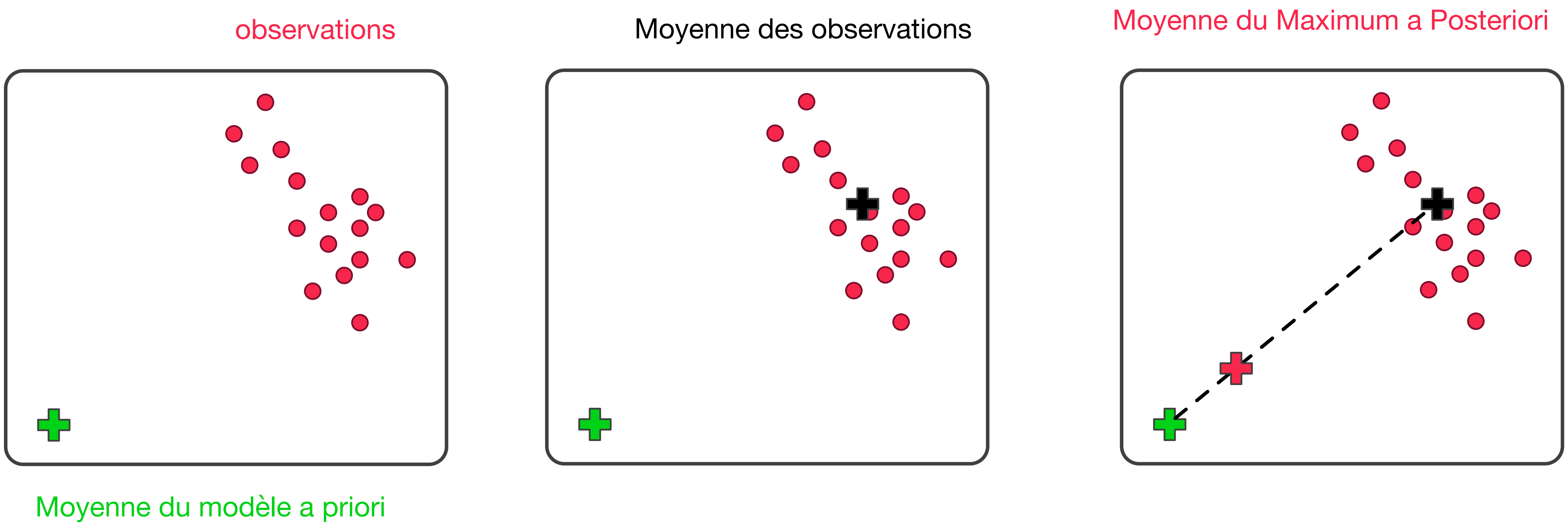
# ADAPTATION DU MODÈLE DU MONDE AU LOCUTEUR

## Maximum a Posteriori (MAP)

- on introduit une information a priori sur les paramètres du modèle à estimer (rôle du modèle du monde)
- On maximise la probabilité des données à posteriori (tout est dans le nom…): $Pr(\theta|x_{1\ldots I})$

# ADAPTATION DU MODÈLE DU MONDE AU LOCUTEUR

- Principe de l'adaptation MAP



# ADAPTATION DU MODÈLE DU MONDE AU LOCUTEUR

## Maximum a Posteriori (MAP)

$$\hat{\theta} = \underset{\theta}{argmax}[Pr(\theta|x_{1...I})]$$

$$\hat{\theta} = \underset{\theta}{argmax}[\frac{Pr(x_{1...I}|\theta)Pr(\theta)}{Pr(x_{1...I})}]$$

Hypothèse d'indépendance des observations
(on a un recouvrement de 60% des fenêtre d'analyse…)

$$\hat{\theta} = \underset{\theta}{argmax}[\frac{\prod_{i=1}^{I} Pr(x_i|\theta)Pr(\theta)}{Pr(x_{1...I})}]$$

# ADAPTATION DU MODÈLE DU MONDE AU LOCUTEUR

## Maximum a Posteriori (MAP)

- À partir du modèle *a priori* (UBM), on calcule les statistiques suffisantes
- Les paramètres obtenus sont une somme pondérée entre le modèle *a priori* (UBM) et les paramètres obtenus par maximum de vraisemblance

# ADAPTATION DU MODÈLE DU MONDE AU LOCUTEUR

## Maximum a Posteriori (MAP)

Formules de mise à jour des paramètres:

$$\hat{w}_c = [\alpha_c \frac{n_c}{I} + (1 - \alpha_c) w_c]\gamma$$

Calculé après pour assurer que la somme des poids vaut 1

$$\hat{\mu}_c = \alpha_c F_c + (1 - \alpha_c)\mu_c$$

$$\hat{\Sigma}_c = [\alpha_c S_c + (1 - \alpha_c)(\Sigma_c + \mu_c \mu_c^T] - \hat{\mu}_c \hat{\mu}_c^T$$

# ADAPTATION DU MODÈLE DU MONDE AU LOCUTEUR

## Maximum a Posteriori (MAP)

$$\hat{w}_c = [\alpha_c \frac{n_c}{I} + (1 - \alpha_c) w_c]\gamma$$

$$\alpha_c = \frac{n_c}{n_c + r}$$

Relevance Factor

$$\hat{\mu_c} = \alpha_c F_c + (1 - \alpha_c)\mu_c$$

$$\hat{\Sigma}_c = [\alpha_c S_c + (1 - \alpha_c)(\Sigma_c + \mu_c \mu_c^T] - \hat{\mu}_c \hat{\mu}_c^T$$

# ADAPTATION DU MODÈLE DU MONDE AU LOCUTEUR

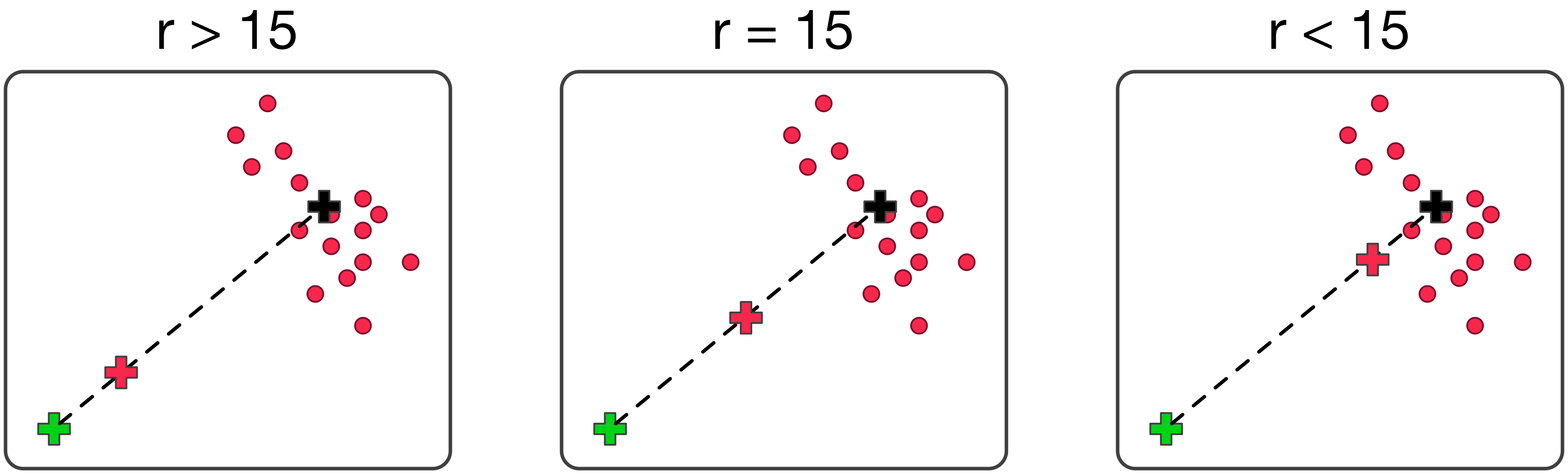
## Maximum a Posteriori (MAP)

Interprétation du relèvent factor « r »: lorsque le nombre d'observations attribuées à la Gaussienne « c » est égal à « r », le maximum a posteriori se trouve au milieu entre le maximum de vraisemblance et l'a priori.

$$\alpha_c = \frac{n_c}{n_c + r}$$

# ADAPTATION DU MODÈLE DU MONDE AU LOCUTEUR

- Interprétation du relevante factor



# ADAPTATION DU MODÈLE DU MONDE AU LOCUTEUR

- En pratique on n'adapte **que les moyennes**
- On a rarement assez de données pour adapter la covariance (même diagonale)
- Adapter les poids apporte très peu en reconnaissance du locuteur ou des langues

# ADAPTATION DU MODÈLE DU MONDE AU LOCUTEUR

- On adapte seulement les moyennes
- Les locuteurs sont donc caractérisés par les vecteurs de moyennes des distributions

Ne pas adapter les variances simplifie les calculs, notamment pour les rapports de vraisemblance

# LE MODÈLE DU MONDE COMME HYPOTHÈSE ALTERNATIVE

Deuxième motivation du modèle du monde:
retour sur le rapport de vraisemblance, comment modéliser l'hypothèse négative?

# LE MODÈLE DU MONDE COMME HYPOTHÈSE ALTERNATIVE

On considère un rapport d'hypothèses: le rapport de vraisemblance:

$$\frac{P(o|H_0)}{P(o|H_1)}$$

Où H0 est l'hypothèse selon laquelle

« o » a été produite par le locuteur cible

et H1 est l'hypothèse selon laquelle

« o » n'a pas été produite par le locuteur cible

# LE MODÈLE DU MONDE COMME HYPOTHÈSE ALTERNATIVE

- On a vu comment modéliser un locuteur avec un GMM
- Comment modéliser « non un locuteur »?
- Le modèle du monde est utilisé pour représenter « tous les locuteurs sauf le locuteur cible »
- Il est impossible d'avoir «  les locuteurs sauf » la cible
- Si on avait tous les locuteur on ferait de l'identification et non de la vérification (meilleures performances)
- **TRÈS IMPORTANT: s'assurer que les clients ne sont pas « dans » le modèle du monde**

# LE MODÈLE DU MONDE COMME HYPOTHÈSE ALTERNATIVE

- Le modèle du monde pour modéliser l'hypothèse négative dans le rapport de vraisemblance
- Permet de donner du poids à ce qui est spécifique à un locuteur et pas commun à tous les locuteurs
- permet une certaine calibration des scores (on compare toujours à la même chose)

# ÉTAPE DE TEST

- Nous avons un modèle du monde (UBM)
- Nous avons adapté un modèle de locuteur à partir des données disponibles (adaptation MAP)
- Comment calculer le score?
  On calcule un log-rapport de vraisemblance (log-likelihood ratio - llk)

# ÉTAPE DE TEST

$$\log Pr(\mathcal{X}|\theta) = \frac{1}{I}\sum_i \log Pr(x_i|\theta)$$

$$\log Pr(\mathcal{X}|\theta) = \frac{1}{I}\sum_i \log \sum_c w_c \mathcal{N}_{x_i}(\mu_c, \Sigma_c)$$

# SUPER-VECTEURS ET SVM

- Les locuteurs ne sont modélisés que par les vecteurs de moyennes des Gaussiennes
- Si on concatène tous les vecteurs de moyenne on obtiens un super-vecteur
- On peut donc représenter un locuteur comme un point dans un « super-espace » tel que:
  10 000 < dimension < 50 000

# OUTILS POUR LA RECONNAISSANCE DU LOCUTEUR / LANGUE

| Outils | Langage |
|---|---|
| ALIZE | C++ |
| BOB/Spear | C++ / Python |
| Kaldi | C++ |
| MSR | Matlab |
| SIDEKIT | Python |

Le Mans Université

# TUTORIAL

- http://lium.univ-lemans.fr/sidekit