On sequence kernels for SVM classification of sets of vectors: *application to speaker verification*

Khalid Daoudi (IRIT-UPS, Toulouse, France)

Major part of the Ph.D. work of Jérôme Louradour

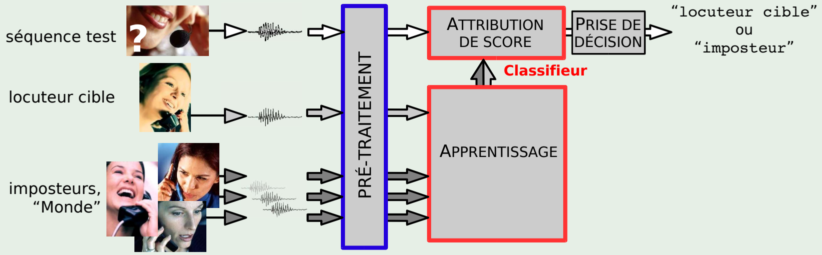
In collaboration with Francis Bach (ARMINES)
within **E-TEAM 11**

Text independent speaker verification

A binary classification task

Determine if a speech sequence has been uttered by a target speaker

Classical approach

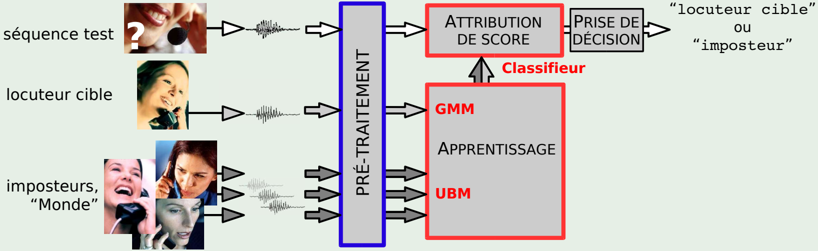


Text independent speaker verification

A binary classification task

Determine if a speech sequence has been uttered by a target speaker

Classical approach : UBM-GMM system



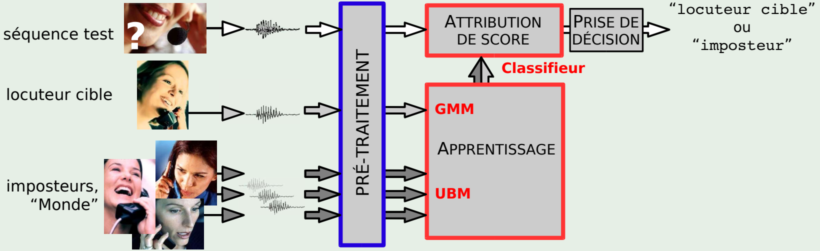
- Probabilistic GMM modeling (generative)

Text independent speaker verification

A binary classification task

Determine if a speech sequence has been uttered by a target speaker

Classical approach : UBM-GMM system



- Probabilistic GMM modeling (generative)

y Motivation : apply SVM, powerful for binary classification

Support Vector Machines (SVM)

+ Good theoretical foundations
+ Well mastered learning algorithms
+ Good results in static data classification

In speech processing ...

– Extension to dynamic data is not easy
– Size of training corpus ⇒ time & memory consuming
– Bad results of SVM when applied at the frame level

Support Vector Machines (SVM)

- + Good theoretical foundations
- + Well mastered learning algorithms
- + Good results in static data classification

In speech processing ...

- – Extension to dynamic data is not easy
- – Size of training corpus ⇒ time & memory consuming
- – Bad results of SVM when applied at the frame level

y Conception and study of **sequence kernels** for speaker verification

Outline

Basics on kernels

- Similarity measure
- Mercer property : symetric, positive definite

$$\Longrightarrow k(\mathrm{x}, \mathrm{y}) = \boldsymbol{\phi}(\mathrm{x})^\top \boldsymbol{\phi}(\mathrm{y})$$

$\boldsymbol{\phi}$: expansion in a *Feature Space* $\mathbb{R}^D$ (dimension $D \leq +\infty$)

Sequence kernels

Three famillies of kernels

1. Mutual Information kernels
   Based on *a priori* data distribution
2. Kernels between probability densities
   Sequence $\mapsto$ distribution
3. Combination of vector kernels
   Function of kernel values between inter(intra)-sequence elements

Sequence kernels

Three famillies of kernels

1. Mutual Information kernels
   Based on *a priori* data distribution
2. Kernels between probability densities
   Sequence $\mapsto$ distribution
3. Combination of vector kernels
   Function of kernel values between inter(intra)-sequence elements

Definition of a sequence

"Variable-length set of acoustic vectors
- same speaker,
- same recording session"

Mutual Information kernels

Exploit the a priori (generative) UBM
which parameters θ_o are esimated on non-labeled data

Fisher kernel [Jaakkola and Haussler, 1998]

- Approximation of a Mutual Information kernel [Seeger, 2002]

$$\kappa(X, Y) = \phi(X)^\top \mathbf{S}^{-1} \phi(Y)$$

- $\phi(X) = \nabla_\theta \log p(X|\theta)|_{\theta=\theta_o}$ (Fisher expantion)
  $\mathbf{S}$: second moments of ϕ (Fisher information matrix)

Mutual Information kernels

Exploit the a priori (generative) UBM
which parameters θ_o are esimated on non-labeled data

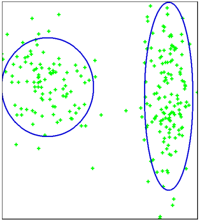
Fisher kernel [Jaakkola and Haussler, 1998]

- Approximation of a Mutual Information kernel [Seeger, 2002]

$$\kappa(X, Y) = \phi(X)^\top \mathbf{S}^{-1} \phi(Y)$$

- $\phi(X) = \nabla_\theta \log p(X|\theta)|_{\theta=\theta_o}$ (Fisher expantion)

  $\mathbf{S}$: second moments of ϕ (Fisher information matrix)



2 Gaussians GMM

Mutual Information kernels

Exploit the a priori (generative) UBM
which parameters θ_o are esimated on non-labeled data

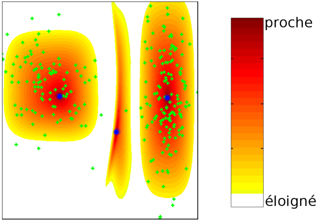
Fisher kernel [Jaakkola and Haussler, 1998]

- Approximation of a Mutual Information kernel [Seeger, 2002]

$$\kappa(X, Y) = \phi(X)^\top \mathbf{S}^{-1} \phi(Y)$$

- $\phi(X) = \nabla_\theta \log p(X|\theta)|_{\theta=\theta_o}$ (Fisher expantion)

  S : second moments of ϕ (Fisher information matrix)



Distance between Fisher expansions

Kernels between probability densities

$$\mathrm{X}, \mathrm{Y} \xrightarrow[\text{learning}]{\text{Generative}} p_{\mathrm{X}}, p_{\mathrm{Y}}$$

probability product kernels [Jebara and Kondor, 2003]

- $$\kappa(\mathrm{X}, \mathrm{Y}) = \int p_{\mathrm{X}}(\mathrm{z})^{q} p_{\mathrm{Y}}(\mathrm{z})^{q}\, \mathrm{dz}$$
- Analytic form a GMM with degree $q = 1$ **[Lyu, 2005]**
- "spherical" normalization for more robustness :
$$\mathring{\kappa}(\mathrm{X},\mathrm{Y}) = \frac{\kappa(\mathrm{X},\mathrm{Y})}{\sqrt{\kappa(\mathrm{X},\mathrm{X})\kappa(\mathrm{Y},\mathrm{Y})}}$$

Exponential embedding of divergences

- Analogous of the Gaussien kernel : $\kappa(\mathrm{X}, \mathrm{Y}) = e^{-\frac{\mathcal{D}(p_{\mathrm{X}}, p_{\mathrm{Y}})^2}{2\rho^2}}$
- $\mathcal{D}$: Distance between GMMs, approximation of KL divergence **[Do, 2003]**

  y "supervectors GMM" Approach **[Campbell et al., 2006]**

Combination of vector kernels

Sequences of vectors $\begin{array}{l} \mathrm{X} = \{\, \mathrm{x}_t \mid t = 1 \cdots T_\mathrm{X} \} \\ \mathrm{Y} = \{\, \mathrm{y}_{t'} \mid t' = 1 \cdots T_\mathrm{Y} \} \end{array}$

Similarity between vectors		Similarity between sets of vectors
$k(\mathrm{x},\mathrm{y}) = \boldsymbol{\phi}(\mathrm{x})^\top \boldsymbol{\phi}(\mathrm{y})$	w	$\kappa(\mathrm{X},\mathrm{Y})$
Mercer kernel		*function of* $k(\mathrm{x}_t,\mathrm{y}_{t'})$, $k(\mathrm{x}_t,\mathrm{x}_{t'})$, $k(\mathrm{y}_t,\mathrm{y}_{t'})$, ...

Simple example :

$$\kappa(\mathrm{X},\mathrm{Y}) = \frac{1}{T_\mathrm{X} T_\mathrm{Y}} \sum_{t=1}^{T_\mathrm{X}} \sum_{t'=1}^{T_\mathrm{Y}} k(\mathrm{x}_t,\mathrm{y}_{t'})$$

complexity $O(T^2)$ for *each* kernel computation

Combination of vector kernels

GLDS kernel [Campbell, 2001]

$$\kappa(\mathrm{X},\mathrm{Y}) = \left(\frac{1}{T_\mathrm{X}}\sum_t \boldsymbol{\phi}_q(\mathrm{x}_t)\right)^\top \mathbf{S}_\mathrm{B}{}^{-1}\left(\frac{1}{T_\mathrm{Y}}\sum_{t'} \boldsymbol{\phi}_q(\mathrm{y}_{t'})\right)$$

$\boldsymbol{\phi}_q$: Polynomial expansion ($\mathbb{R}^d \to \mathbb{R}^D$)
$D = \frac{(q+d)!}{q!d!}$ monomes of degree $\leq q$

$\mathbf{S}_\mathrm{B}$: Matrix of second moments of $\boldsymbol{\phi}_q$ estimated on B

- Normalization by $\mathbf{S}_B$:
  - Kernel $\sim$ scoring on X a discr. model learned on Y
  - Same amplitude for each component of the *Feature Space*
- Explicit expansion of data :
  - \+ high efficiency during test (linear SVM model)
  - − Impossible to use expansions $\boldsymbol{\phi}_q$ of high or infinite dimension (in practice, maxi degree $q = 3$)

Combination of vector kernels

GLDS kernel [Campbell, 2001]

$$\kappa(\mathrm{X}, \mathrm{Y}) = \left(\frac{1}{T_\mathrm{X}} \sum_t \boldsymbol{\phi}_q(\mathrm{x}_t) \right)^\top \mathbf{S}_\mathrm{B}{}^{-1} \left(\frac{1}{T_\mathrm{Y}} \sum_{t'} \boldsymbol{\phi}_q(\mathrm{y}_{t'}) \right)$$

$\boldsymbol{\phi}_q$: Polynomial expansion ($\mathbb{R}^d \to \mathbb{R}^D$)
$D = \frac{(q+d)!}{q!d!}$ monomes of degree $\leq q$
$\mathbf{S}_\mathrm{B}$: Matrix of second moments of $\boldsymbol{\phi}_q$ estimated on B

- Normalization by $\mathbf{S}_B$:
  - Kernel $\sim$ scoring on X a discr. model learned on Y
  - Same amplitude for each component of the *Feature Space*
- Explicit expansion of data :
  - \+ high efficiency during test (linear SVM model)
  - − Impossible to use expansions $\boldsymbol{\phi}_q$ of high or infinite dimension (in practice, maxi degree $q = 3$)

Outline

Extension of the GLDS kernel : FSNS kernels

FSNS kernels (*Feature Space Normalized Sequence kernels*)

$$\kappa(\mathrm{X},\mathrm{Y}) = \left(\frac{1}{T_{\mathrm{X}}}\sum_t \boldsymbol{\phi}(\mathrm{x}_t)\right)^{\top} \left(\mathbf{S}_{\mathrm{B}} + \varepsilon \mathbf{I}\right)^{-1} \left(\frac{1}{T_{\mathrm{Y}}}\sum_{t'} \boldsymbol{\phi}(\mathrm{y}_{t'})\right)$$

- $\boldsymbol{\phi}$: any Mercer expansion
- $\mathbf{S}_{\mathrm{B}}$: second moments matrix of $\boldsymbol{\phi}$
- Regularization $\varepsilon > 0$: necessary if $\boldsymbol{\phi}$ is of high dimension

y Objectif : $\begin{cases} \text{avoid to compute } \boldsymbol{\phi} \\ \text{rewrite using the Mercer kernel } k = \boldsymbol{\phi}^{\top}\boldsymbol{\phi} \end{cases}$

Extension of the GLDS kernel : FSNS kernels

FSNS kernels (*Feature Space Normalized Sequence kernels*)

$$\kappa(\mathrm{X},\mathrm{Y}) = \left(\frac{1}{T_\mathrm{X}}\sum_t \phi(\mathrm{x}_t)\right)^\top \left(\mathbf{S}_\mathrm{B} + \varepsilon\mathbf{I}\right)^{-1} \left(\frac{1}{T_\mathrm{Y}}\sum_{t'} \phi(\mathrm{y}_{t'})\right)$$

- ϕ : any Mercer expansion
- $\mathbf{S}_\mathrm{B}$: second moments matrix of ϕ
- Regularization $\varepsilon > 0$: necessary if ϕ is of high dimension

y Objectif : { avoid to compute ϕ ; rewrite using the Mercer kernel $k = \phi^\top\phi$ }

FSMS kernels (*Feature Space Mahalanobis Sequence kernels*)

$$\kappa(\mathrm{X},\mathrm{Y}) = \left(\frac{1}{T_\mathrm{X}}\sum_t \phi(\mathrm{x}_t)\right)^\top \left(\mathbf{\Sigma}_\mathrm{B} + \varepsilon\mathbf{I}\right)^{-1} \left(\frac{1}{T_\mathrm{Y}}\sum_{t'} \phi(\mathrm{y}_{t'})\right)$$

- $\mathbf{\Sigma}_\mathrm{B}$: covariance matrix of ϕ
- SVM are invariant to translations in the *Feature Space*
  $\Rightarrow$ same as FSNS with centring of ϕ
- kernel $\sim$ KL divergence between Gaussians in the *Feature Space*

Dual form of FSNS kernels

Training (Background) data : $\mathrm{B} = \{\ \mathrm{b}_i \mid i = 1 \cdots N\}$

↙ ↘

Gram matrix

$$\mathbf{K} = \begin{bmatrix} k(\mathrm{b}_1,\mathrm{b}_1) & \cdots & k(\mathrm{b}_1,\mathrm{b}_N) \\ \vdots & k(\mathrm{b}_i,\mathrm{b}_j) & \vdots \\ k(\mathrm{b}_N,\mathrm{b}_1) & \cdots & k(\mathrm{b}_N,\mathrm{b}_N) \end{bmatrix}$$

Empirical map

$$\psi_{\mathrm{B}}(\mathrm{x}) = \begin{bmatrix} k(\mathrm{b}_1,\mathrm{x}) \\ \vdots \\ k(\mathrm{b}_N,\mathrm{x}) \end{bmatrix}$$

Proposition (without regularization, $\varepsilon = 0$)

$$\kappa(\mathrm{X},\mathrm{Y}) = \left(\frac{1}{T_{\mathrm{X}}}\sum_t \phi(\mathrm{x}_t)\right)^{\top} \mathbf{S}_{\mathrm{B}}{}^{-1} \left(\frac{1}{T_{\mathrm{Y}}}\sum_{t'} \phi(\mathrm{y}_{t'})\right)$$

$$= \boxed{\left(\frac{1}{T_{\mathrm{X}}}\sum_t \psi_{\mathrm{B}}(\mathrm{x}_t)\right)^{\top} \left(\frac{1}{N}\mathbf{K}^2\right)^{-1} \left(\frac{1}{T_{\mathrm{Y}}}\sum_{t'} \psi_{\mathrm{B}}(\mathrm{y}_{t'})\right)}$$

Dual form of FSNS kernels

Training (Background) data : $\mathrm{B} = \{ \mathrm{b}_i \mid i = 1 \cdots N\}$

$\swarrow \quad \searrow$

Gram matrix **Empirical map**

$$\mathbf{K} = \begin{bmatrix} k(\mathrm{b}_1,\mathrm{b}_1) & \cdots & k(\mathrm{b}_1,\mathrm{b}_N) \\ \vdots & k(\mathrm{b}_i,\mathrm{b}_j) & \vdots \\ k(\mathrm{b}_N,\mathrm{b}_1) & \cdots & k(\mathrm{b}_N,\mathrm{b}_N) \end{bmatrix} \qquad \psi_\mathrm{B}(\mathrm{x}) = \begin{bmatrix} k(\mathrm{b}_1,\mathrm{x}) \\ \vdots \\ k(\mathrm{b}_N,\mathrm{x}) \end{bmatrix}$$

Proposition

$$\begin{aligned} \kappa(\mathrm{X},\mathrm{Y}) &= \left(\frac{1}{T_\mathrm{X}}\sum_t \phi(\mathrm{x}_t)\right)^\top \left(\mathbf{S}_\mathrm{B} + \varepsilon \mathbf{I}\right)^{-1} \left(\frac{1}{T_\mathrm{Y}}\sum_{t'} \phi(\mathrm{y}_{t'})\right) \\ &= \boxed{\left(\frac{1}{T_\mathrm{X}}\sum_t \psi_\mathrm{B}(\mathrm{x}_t)\right)^\top \left(\frac{1}{N}\mathbf{K}^2 + \varepsilon \mathbf{K}\right)^{-1} \left(\frac{1}{T_\mathrm{Y}}\sum_{t'} \psi_\mathrm{B}(\mathrm{y}_{t'})\right)} \end{aligned}$$

- Hypothesis : the $\phi(\mathrm{b}_i)$ span the training "$\phi(\mathrm{x})$"

Dual form of FSNS kernels

Training (Background) data : $\mathrm{B} = \{ \mathrm{b}_i \mid i = 1 \cdots N\}$

Gram matrix

$$\mathbf{K} = \begin{bmatrix} k(\mathrm{b}_1,\mathrm{b}_1) & \cdots & k(\mathrm{b}_1,\mathrm{b}_N) \\ \vdots & k(\mathrm{b}_i,\mathrm{b}_j) & \vdots \\ k(\mathrm{b}_N,\mathrm{b}_1) & \cdots & k(\mathrm{b}_N,\mathrm{b}_N) \end{bmatrix}$$

Empirical map

$$\psi_{\mathrm{B}}(\mathrm{x}) = \begin{bmatrix} k(\mathrm{b}_1,\mathrm{x}) \\ \vdots \\ k(\mathrm{b}_N,\mathrm{x}) \end{bmatrix}$$

Proposition (with centering)

$$\kappa(\mathrm{X},\mathrm{Y}) = \left(\frac{1}{T_{\mathrm{X}}}\sum_t \phi(\mathrm{x}_t)\right)^{\top} \left(\boldsymbol{\Sigma}_{\mathrm{B}} + \varepsilon \mathbf{I}\right)^{-1} \left(\frac{1}{T_{\mathrm{Y}}}\sum_{t'} \phi(\mathrm{y}_{t'})\right)$$

$$= \boxed{\left(\frac{1}{T_{\mathrm{X}}}\sum_t \psi_{\mathrm{B}}(\mathrm{x}_t)\right)^{\top} \left(\frac{1}{N}\mathbf{K}\Pi\mathbf{K} + \varepsilon \mathbf{K}\right)^{-1} \left(\frac{1}{T_{\mathrm{Y}}}\sum_{t'} \psi_{\mathrm{B}}(\mathrm{y}_{t'})\right)}$$

- Hypothesis : the $\phi(\mathrm{b}_i)$ span the training "$\phi(\mathrm{x})$"
- Centering : $\Pi = \mathbf{I} - \frac{1}{N}\mathbf{1}$ (instead of $\mathbf{I}$)

Computational complexity

Dot product of normalized expansions

$$\kappa(X,Y) = \overline{\phi}(X)^{\top} \mathbf{M}_B\, \overline{\phi}(Y) = \langle\, \mathbf{U}\overline{\phi}(X)\,,\, \mathbf{U}\overline{\phi}(Y)\,\rangle \quad (1)$$

$$= \overline{\psi}_B(X)^{\top} \mathbf{R}_B\, \overline{\psi}_B(Y) = \langle\, \mathbf{V}\overline{\psi}_B(X)\,,\, \mathbf{V}\overline{\psi}_B(Y)\,\rangle \quad (2)$$

	form (1)	form (2)
Pre-computation of $\mathbf{U}/\mathbf{V}$	$O(D^3)$	$O(N^3)$
Sequence expansion $\mathbf{U}\overline{\phi}/\mathbf{V}\overline{\psi}_B$	$O(TD^2)$	$O(TN^2)$
Dot product comput.	$O(D)$	$O(N)$

D : dimension of the *Feature Space* (size of ϕ)
N : number of background vectors (size of ψ)

+ Possibility to use expansions ϕ of infinite dim (Gaussian kernel)
− Complexity problem for large databases

Computational complexity

Dot product of normalized expansions

$$\kappa(X,Y) = \overline{\phi}(X)^\top \mathbf{M}_B\, \overline{\phi}(Y) = \langle\, \mathbf{U}\overline{\phi}(X)\,,\, \mathbf{U}\overline{\phi}(Y)\,\rangle \quad (1)$$
$$= \overline{\psi}_B(X)^\top \mathbf{R}_B\, \overline{\psi}_B(Y) = \langle\, \mathbf{V}\overline{\psi}_B(X)\,,\, \mathbf{V}\overline{\psi}_B(Y)\,\rangle \quad (2)$$

	form (1)	form (2)
Pre-computation of $\mathbf{U}/\mathbf{V}$	$O(D^3)$	$O(N^3)$
Sequence expansion $\mathbf{U}\overline{\phi}/\mathbf{V}\overline{\psi}_B$	$O(TD^2)$	$O(TN^2)$
Dot product comput.	$O(D)$	$O(N)$

D : dimension of the *Feature Space* (size of ϕ)
N : number of background vectors (size of ψ)

\+ Possibility to use expansions ϕ of infinite dim (Gaussian kernel)

– Complexity problem for large databases

y Objective : find an appropriate approximation

Kernel Approximation

Goal

1. Reduce the size of the empirical map ψ
2. Keep a maximum of information

ICD : Incomplete Cholesky Decomposition [Fine and Scheinberg, 2001]

1. Selection of a sub-population of backround vectors :

$$\begin{array}{rl} \textit{codebook} & \mathrm{C} = \{ \mathrm{b}_{p_1} \cdots \mathrm{b}_{p_i} \cdots \mathrm{b}_{p_m} \} \subset \mathrm{B} \\ \textit{index} & \mathrm{I} = \{ p_1 \cdots p_i \cdots p_m \} \subset \{1 \cdots N\} \end{array} \qquad \text{(taille } m\text{)}$$

2. Low-rank approximation of the Gram matrix :

$$\mathbf{K} \approx \mathbf{L}_\mathrm{I} = \mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^\top \qquad \text{(rang } m\text{)}$$

$$\min_{\mathrm{I}} \mathrm{tr}\left(\mathbf{K} - \mathbf{L}_\mathrm{I}\right) \equiv \min_{\mathrm{C}} \sum \|\phi(\mathrm{b}_i) - \phi_\mathrm{C}(\mathrm{b}_i)\|^2$$

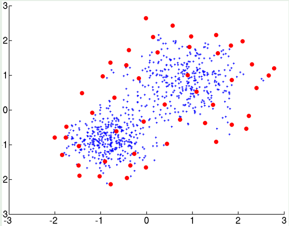
\+ CPU and memory

Kernel Approximation

Goal

1. Reduce the size of the empirical map ψ
2. Keep a maximum of information

ICD : Incomplete Cholesky Decomposition



Training data
Codebook

ICD, Gaussian kernel

Approximate Form of FSNS kernels

Proposition

ICD

$$\mathbf{K} \quad \mathrm{w} \quad \mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^{\top}$$

$$\kappa(\mathrm{X},\mathrm{Y}) = \overline{\psi}_{\mathrm{B}}(\mathrm{X})^{\top}\mathbf{R}_{\mathrm{B}}\,\overline{\psi}_{\mathrm{B}}(\mathrm{Y}) \quad \mathrm{w} \quad \boxed{\kappa(\mathrm{X},\mathrm{Y}) \approx \overline{\psi}_{\mathrm{C}}(\mathrm{X})^{\top}\mathbf{R}_{\mathrm{B}\times\mathrm{C}}\,\overline{\psi}_{\mathrm{C}}(\mathrm{Y})}$$

- Expansion of size $m \ll N$:

$$\overline{\psi}_{\mathrm{C}}(\mathrm{X}) = \frac{1}{T_{\mathrm{X}}}\sum_t \begin{bmatrix} k(\mathrm{b}_{p_1},\mathrm{x}_t) \\ \vdots \\ k(\mathrm{b}_{p_m},\mathrm{x}_t) \end{bmatrix}$$

$$\mathbf{R}_{\mathrm{B}\times\mathrm{C}} = \left(\frac{1}{N}\mathbf{K}(:,\mathrm{I})\Pi\mathbf{K}(:,\mathrm{I})^{\top} + \varepsilon\mathbf{K}(\mathrm{I},\mathrm{I})\right)^{-1}$$

- Complexity

	form (1)	form (2)	approx form
Expansion norm.	$O(TD^2)$	$O(TN^2)$	$O(Tm^2)$
Dot product	$O(D)$	$O(N)$	$O(m)$

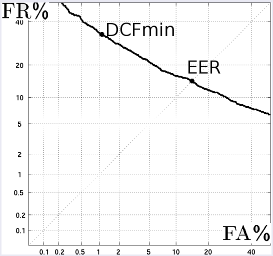
Outline

Data

Speech corpus : `NIST Speaker Recognition Evaluation`

Development : `NIST 2003 and 2004`

- Background corpus `(1/2)`
- Validation corpus `(2/2)`
  - Hyper-parameters of kernels
  - Parameter *C* (SVM leraning)
  - Decision threshold



Evaluation : `NIST 2005`

$\sim$ 18000 tests, 400 target speakers

$$\mathrm{DCF} = \tau_{\mathrm{FR}} P_{\mathrm{loc}} \mathrm{FR}\% + \tau_{\mathrm{FA}} P_{\mathrm{imp}} \mathrm{FA}\%$$

Data

Speech corpus : NIST Speaker Recognition Evaluation

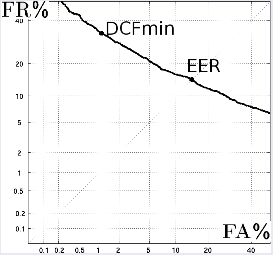
Development : NIST 2003 and 2004

- Background corpus (1/2)
- Validation corpus (2/2)
  - Hyper-parameters of kernels
  - Parameter C (SVM leraning)
  - Decision threshold

Evaluation : NIST 2005

$\sim$ 18000 tests, 400 target speakers

$\mathrm{DCF} = \tau_{\mathrm{FR}} P_{\mathrm{loc}} \mathrm{FR}\% + \tau_{\mathrm{FA}} P_{\mathrm{imp}} \mathrm{FA}\%$

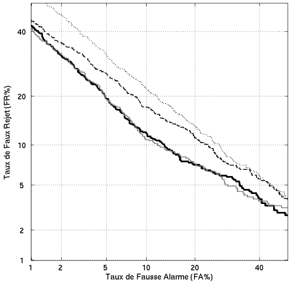


Preprocessing

	SVM	GMM
acoustic vectors	$MFCC+\Delta MFCC+\Delta\log E$	$LFCC+\Delta LFCC+\Delta\log E$
silence suppression	clustering of the energy	
normalization	feature warping	centring-reduction

Development : Centring & Regularization ?

$$\kappa(X,Y) = \overline{\psi_C}(X)^\top \left[\frac{1}{N}\mathbf{K}(:,I)^\top \mathbf{P}\mathbf{K}(:,I) + \varepsilon \mathbf{K}(I,I)\right]^{-1} \overline{\psi_C}(Y)$$

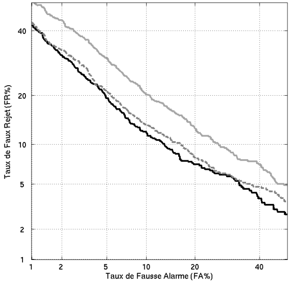


	Centring	Regul.	EER(%)	DCF_{min}
······	$\mathbf{P} = \Pi$	$\varepsilon = 10^{-4}$	15.83	63.5
--	$\mathbf{P} = \Pi$	$\varepsilon = 10^{-6}$	14.11	53.1
— (grey)	$\mathbf{P} = \Pi$	$(\varepsilon = 0)$	**10.62**	**49.3**
— (black)	$(\mathbf{P} = \mathbf{I})$	$(\varepsilon = 0)$	10.95	49.9

- No significant gain with centering
- Regularization degrades

Developement : Choice and tuning of the vector kernel

$$\kappa(\mathrm{X},\mathrm{Y}) = \overline{\psi_{\mathrm{C}}}(\mathrm{X})^{\top} \left[\frac{1}{N}\mathbf{K}(:,\mathrm{I})^{\top}\mathbf{P}\mathbf{K}(:,\mathrm{I}) + \varepsilon\mathbf{K}(\mathrm{I},\mathrm{I})\right]^{-1} \overline{\psi_{\mathrm{C}}}(\mathrm{Y})$$

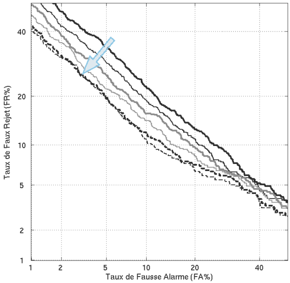


		EER(%)	DCF_{min}
—	$\rho = 2\rho_0$	15.7	60.2
--	$\rho = \rho_0/2$	11.90	50.5
—	$\rho = \rho_0$	**10.95**	**49.9**

- Best results are obtained with a Gaussian kernel $k(\mathbf{x},\mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\rho^2}}$
- ... using a good spread parameter $\rho \approx \rho_0 = \sqrt{d\overline{\sigma}}$

Developement : Size of the codebook

$$\kappa(X,Y) = \overline{\psi_{C}}(X)^{\top} \left[\frac{1}{N}\mathbf{K}(:,I)^{\top}\mathbf{P}\mathbf{K}(:,I) + \varepsilon\mathbf{K}(I,I)\right]^{-1} \overline{\psi_{C}}(Y)$$

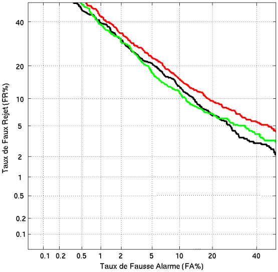


Codebook size		EER(%)	DCF_{min}
—	$m = 300$	15.70	65.0
—	$m = 600$	14.38	61.2
—	$m = 1250$	13.59	57.4
—	$m = 2500$	12.52	53.5
--	$m = 5000$	10.95	49.9
--	$m = 8000$	**10.17**	**48.9**

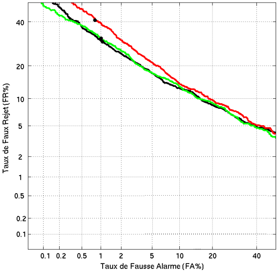
- ☛ Augmenting la codebook size improve the performances
- ☛ . . . until a certain point ($m \sim 5000$)

Evaluation : FSNS kernels vs. Classical approches

Validation corpus



Evaluation corpus



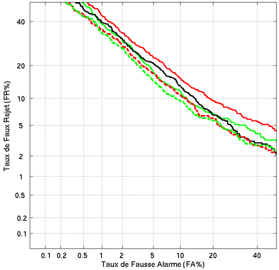
		EER(%)	DCFmin
— (red)	(1) GLDS SVM	12.80	51.9
— (green)	(2) **FSNS SVM**	**10.55**	**47.5**
— (black)	(3) UBM-GMM	11.48	49.1

		EER	DCF
— (red)	(1) GLDS SVM	12.54	48.8
— (green)	(2) **FSNS SVM**	**11.91**	41.6
— (black)	(3) UBM-GMM	12.06	**40.6**

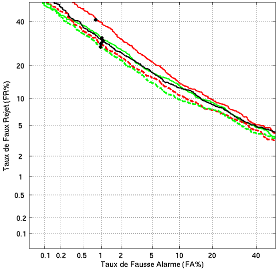
- FSNS kernel : better performances than GLDS kernel
- FSNS-SVM competitive with UBM-GMM

Evaluation : Perspectives of fusion

Validation corpus



Evaluation corpus



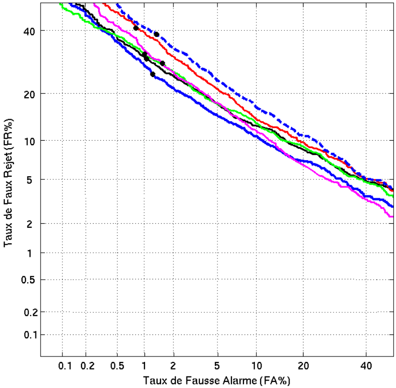
		EER(%)	DCF_{min}
— (red)	(1) GLDS SVM	12.80	51.9
— (green)	(2) **FSNS SVM**	10.55	47.5
— (black)	(3) UBM-GMM	11.48	49.1
	(1+2) fusion	no improvment	
-- (red)	(1+3) fusion	10.32	45.0
-- (green)	(2+3) fusion	**9.50**	**43.5**

		EER	DCF
— (red)	(1) GLDS SVM	12.54	48.8
— (green)	(2) **FSNS SVM**	11.91	41.6
— (black)	(3) UBM-GMM	12.06	40.6
	(1+2) fusion	no improvment	
-- (red)	(1+3) fusion	10.54	38.8
-- (green)	(2+3) fusion	**9.71**	**37.0**

☛ Gain in discriminative / generative fusion

Evaluation : All sequence kernels

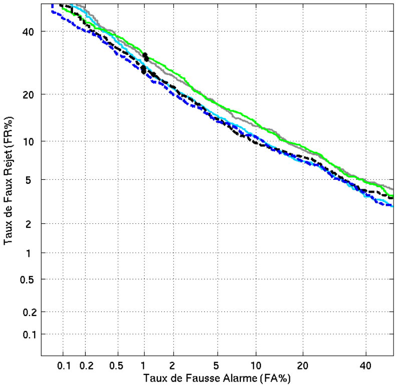
		EER	DCF
-- (blue dashed)	Probability product kernel	13.92	52.1
— (red)	GLDS kernel	12.54	48.8
— (magenta)	Fisher kernel	11.90	44.0
— (green)	**FSNS kernel**	11.91	41.6
— (black)	UBM-GMM (ref)	12.06	40.6
— (blue)	supervectors GMM	**10.40**	**37.7**



☛ Best results : exploiting GMM in SVM

Evaluation : Fusion

		EER	DCF
—	(1) **FSNS kernel**	11.91	41.6
—	(2) UBM-GMM	12.06	40.6
—	(3) Supervectors GMM	10.40	37.7
	(2+3) fusion	no improvment	
--	(1+2) fusion	**9.71**	37.0
--	(1+3) fusion	10.28	**36.1**



☛ Gain in GMM / SVM fusion

Conclusions and future work

Theoretical and experimental exploration of kernels between variable-length sets

- A lot of possible kernels
- A novel kernel (FSNS)
- Experimental comparison on NIST SRE
- Best performance : fusion FSNS/ GMM supervectors

Several ways to improve SVM for speaker verification

- Model adaptation
- String kernels for high-level features (prosody...)
- How to handle SVM scores ?