



Mémoire de MAGISTER



Optimisation des Machines à Vecteurs de Support Multiclasses par l'Algorithme Génétique

Présenté par

M^{me} Fréha MEZZOUDJ

Plan de travail

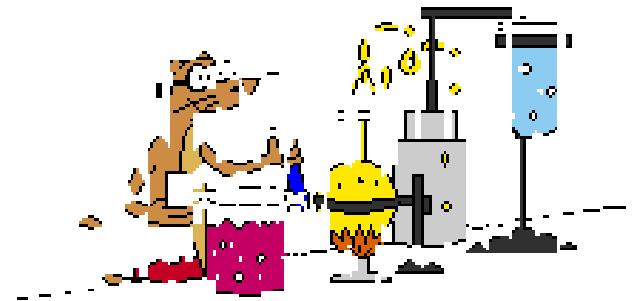
1. Introduction
2. Apprentissage Statistique
3. Machines à Vecteurs de Support (SVM)
4. Algorithmes Génétiques (AG)
5. Parole (RAP)
6. Optimisations proposées (IS, FS)
7. Résultats expérimentaux
8. Conclusions & perspectives

Introduction

- Utilisation des **SVM** >> □ **Résolution d'un QP** ;
- **Utilisation de grandes DB** □ Complexité de calcul ;
- Optimisation possible ?
- Influence de la taille des BD sur le classifieur ?

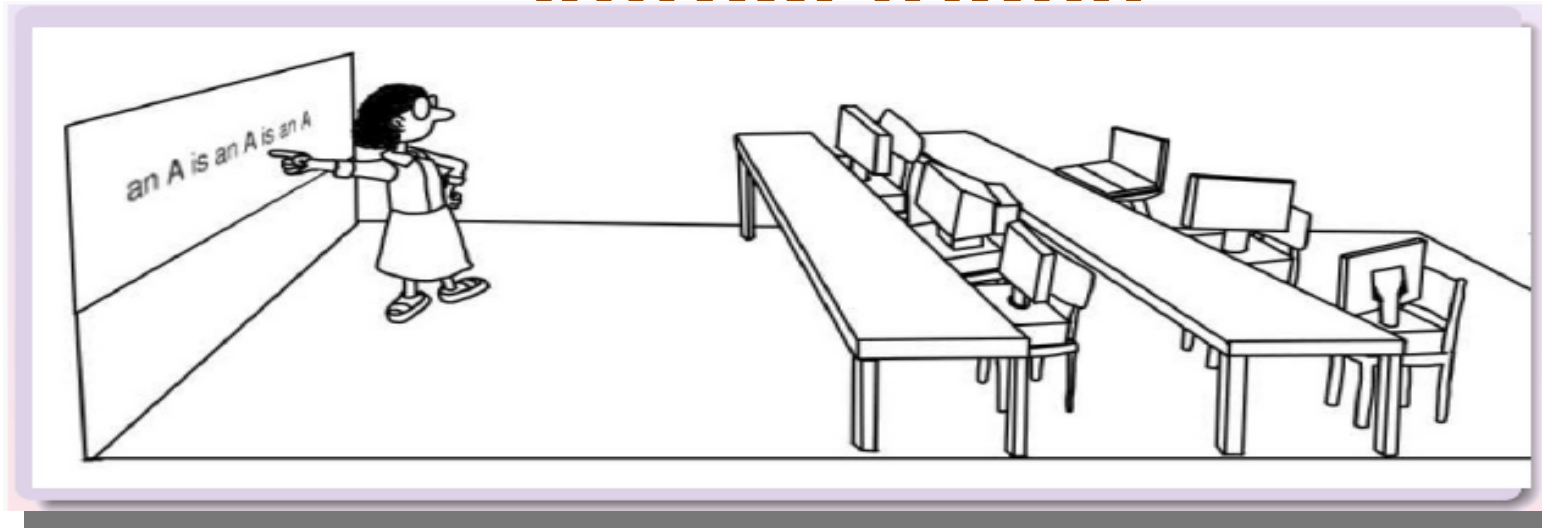


Apprentissage e Statistique



Apprentissage

Automatique



- L'**apprentissage automatique** est une tentative de reproduire la faculté d'apprentissage des humains dans des systèmes artificiels.
- Concevoir des algorithmes capables, à partir d'un nombre d'exemples d'**expériences passées**, d'en assimiler la nature afin de pouvoir appliquer ce qu'ils ont ainsi appris aux **cas futurs**.

Apprentissage Automatique & Statistiques

Apprentissag

e Automatique

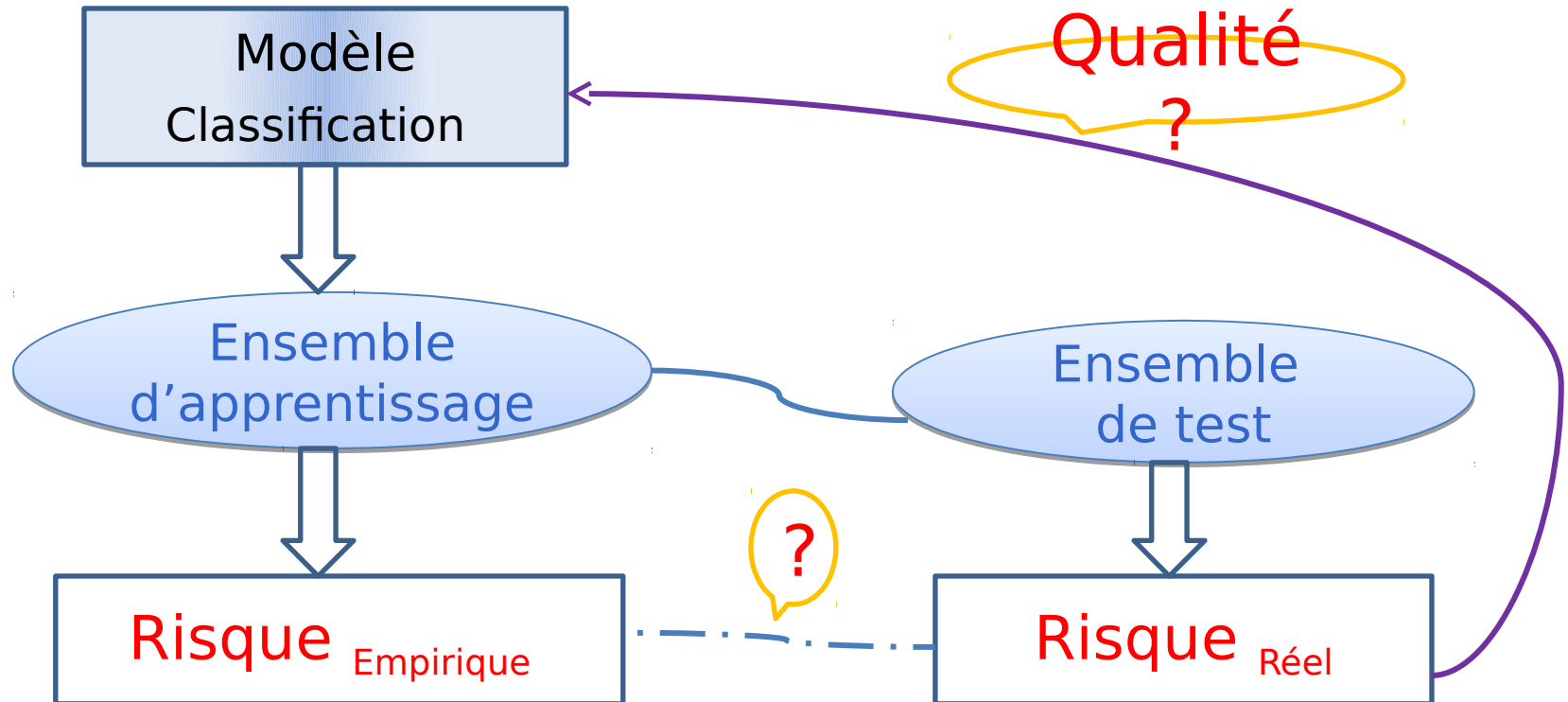
Statistiques

mathématique



**Apprentissag
e Statistique**

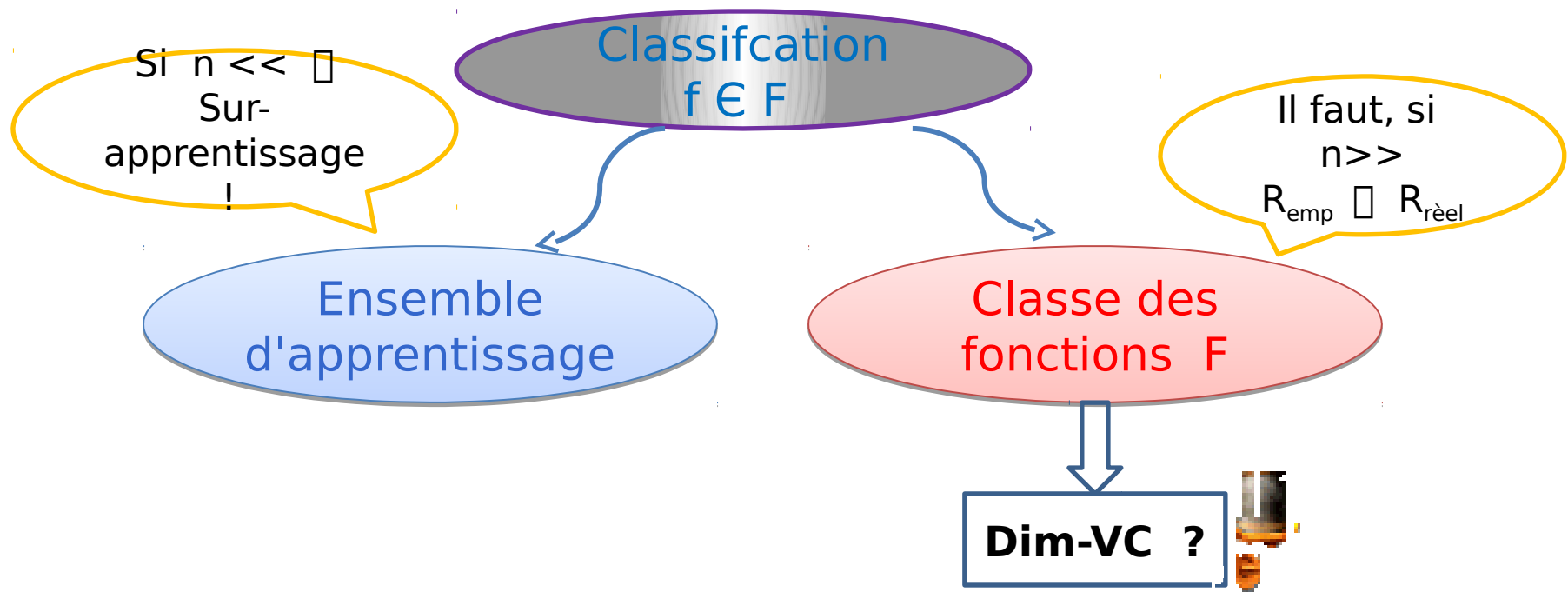
Minimisation du **Risque Empirique**



Lorsqu'on choisit une hypothèse minimisant le **Risque Empirique**, est ce qu' on minimise aussi le **Risque Réel** ?

□ ce critère de sélection d'hypothèse est le **Principe de Minimisation du Risque Empirique**

Minimisation du Risque Empirique

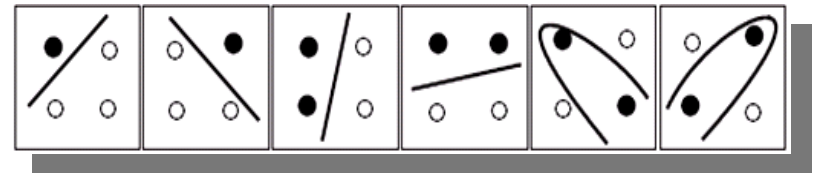
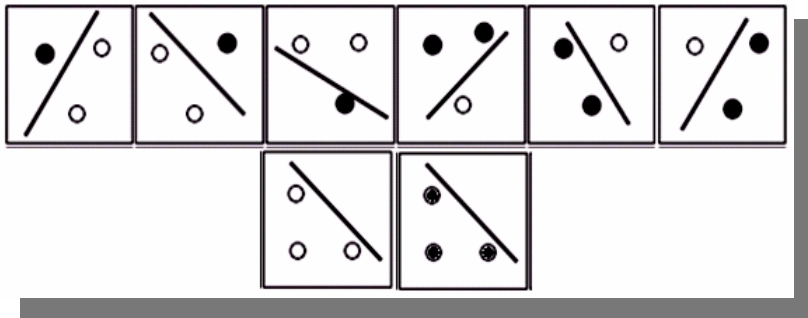


- **Vapnik** & **Chervonenkis** ont introduit un terme de **régularisation** pour mesurer la complexité des fonctions de la classe F dite :

la **dimension de Vapnik-Chervonenkis** (ou la capacité h de F).

Dim-VC ?

Une illustration **pas** une généralisation !



➤ Pour l'ensemble des droites sur \mathbb{R}^2 , **Dim-VC** = 3.

- Mesure la **richesse** (**complexité**) d'un ensemble de fonction ;
- Liée au **nombre maximum** de point pouvant être correctement **séparés** quelque soit leur étiquetage.
- Pour que le principe de **ERM** ($R_{\text{emp}} \square R_{\text{réel}}(f)$) soit consistant il suffit que l'ensemble des fonctions que le système d'apprentissage est capable d'implémenter possède une **dim-VC finie**.

Minimisation du **Risque Structurel**

Vapnik a introduit un Théorème qui permet **borner** le **$R_{\text{réel}}$** (f) tout en **minimisant** le **R_{emp}** :



Théorème

Si F possède une dimension de Vapnik-Chervonenkis finie h , tel que $n > h$ et $\tau = (\sup_F L - \inf_F L)$ alors pour un $\eta > 0$ donné, on a :

$$P \left\{ \sup_{f \in F} |R_{\text{emp}}(f, A_n) - R_{\text{réel}}(f)| \geq \tau \sqrt{\frac{h \left(\log \frac{2n}{h} + 1 \right) - \log \frac{\eta}{4}}{n}} \right\} \leq \eta$$

Minimisation du Risque Structurel

Après quelques déductions, on obtient :

$$\mathbf{R}_{\text{réel}}(f_{A_n}^*) \leq \mathbf{R}_{\text{emp}}(f_{A_n}^*, A_n) + \tau \sqrt{\frac{h \left(\log 2 \frac{n}{h} + 1 \right) - \log \frac{\eta}{4}}{n}}$$

✓ $\mathbf{R}_{\text{emp}}(f_{A_n}^*, A_n)$: \mathbf{R}_{emp} (biais) ;

✓ $\tau \sqrt{\frac{h \left(\log 2 \frac{n}{h} + 1 \right) - \log \frac{\eta}{4}}{n}}$: Largeur de l'intervalle de confiance (variance).

Le second membre de l'inégalité est composé de deux termes :

le 1^{er} terme est le minimum du \mathbf{R}_{emp} sur l'ensemble d'apprentissage A_n

& le 2^{eme} terme une quantité, dépendante de (**n le nombre d'exemples d'apprentissage / h dim-VC du modèle**);

➤ Le principe de **SRM** qu'a introduit **Vapnik** est donc basé sur la **minimisation conjointe** des **deux causes d'erreur** : le \mathbf{R}_{emp} & l'**intervalle de confiance** en utilisant la **dim-VC** comme variable de contrôle.

$$R(T_i) \leq R_{\text{emp}}(T_i) + \frac{\ln N - \ln \eta}{l} \left(1 + \sqrt{1 + \frac{2R_{\text{emp}}(T_i)l}{\ln N - \ln \eta}} \right)$$

ALL YOUR
BAYES ARE
BELONG
TO VS



Machines à Vecteurs de Support (SVMs)



Machines à Vecteurs de Support (SVM)

> SVM Linéaire

- Données linéairement séparable (Sans erreurs)
- Données non linéairement séparable (Avec erreurs)

> SVM non Linéaire

> SVM Multiclasse

SVM Linéaire (Sans erreurs)

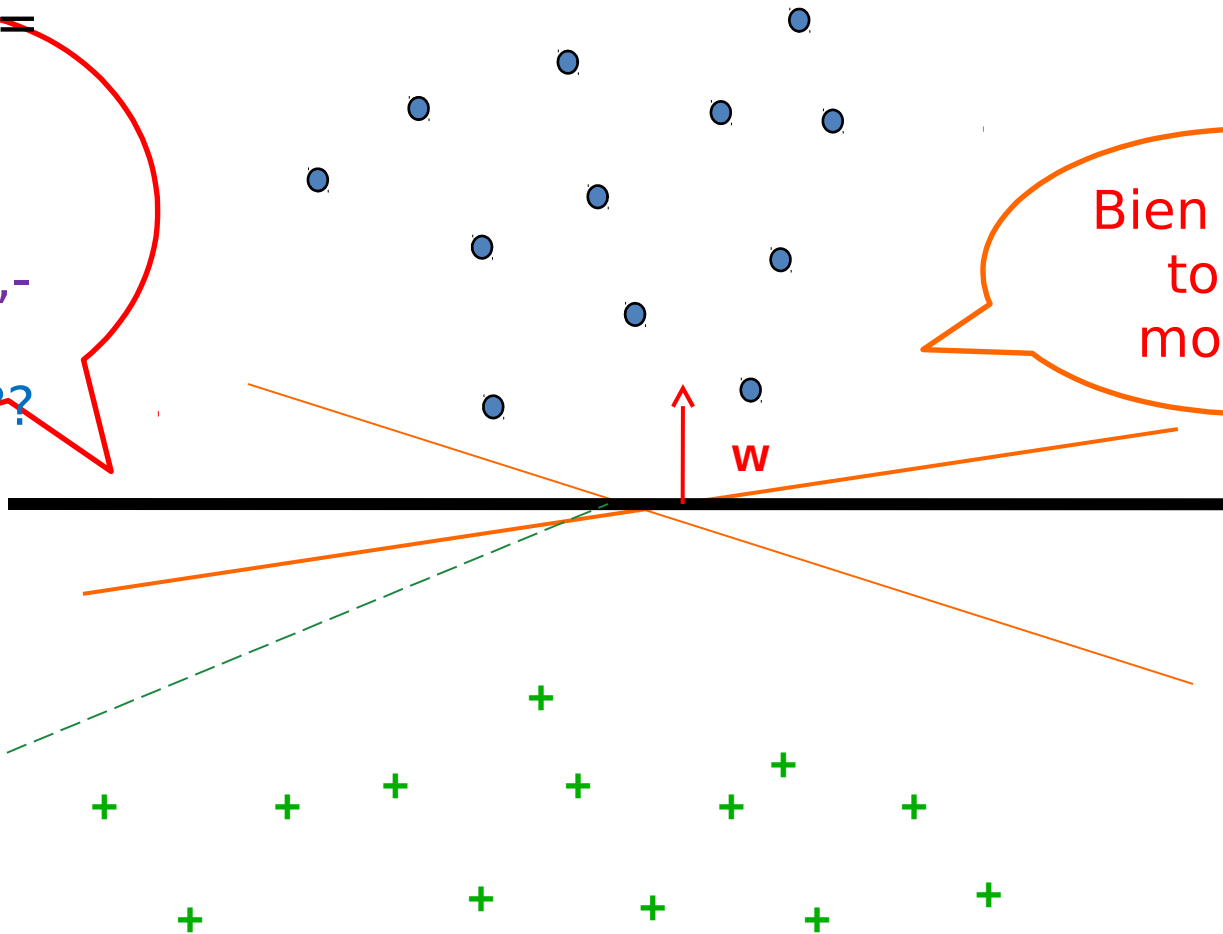
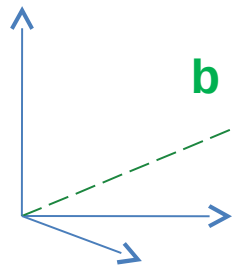
$$wx + b =$$

0

$x \in \mathbb{R}^m$,
 $y \in \{+1, -1\}$

$(w, b) ???$

Bien classer
tout le
monde ?



Origin
e

SVM Linéaire (Sans erreurs)

$$w \cdot x_1 = +1$$

$$w \cdot x_2 = -1$$

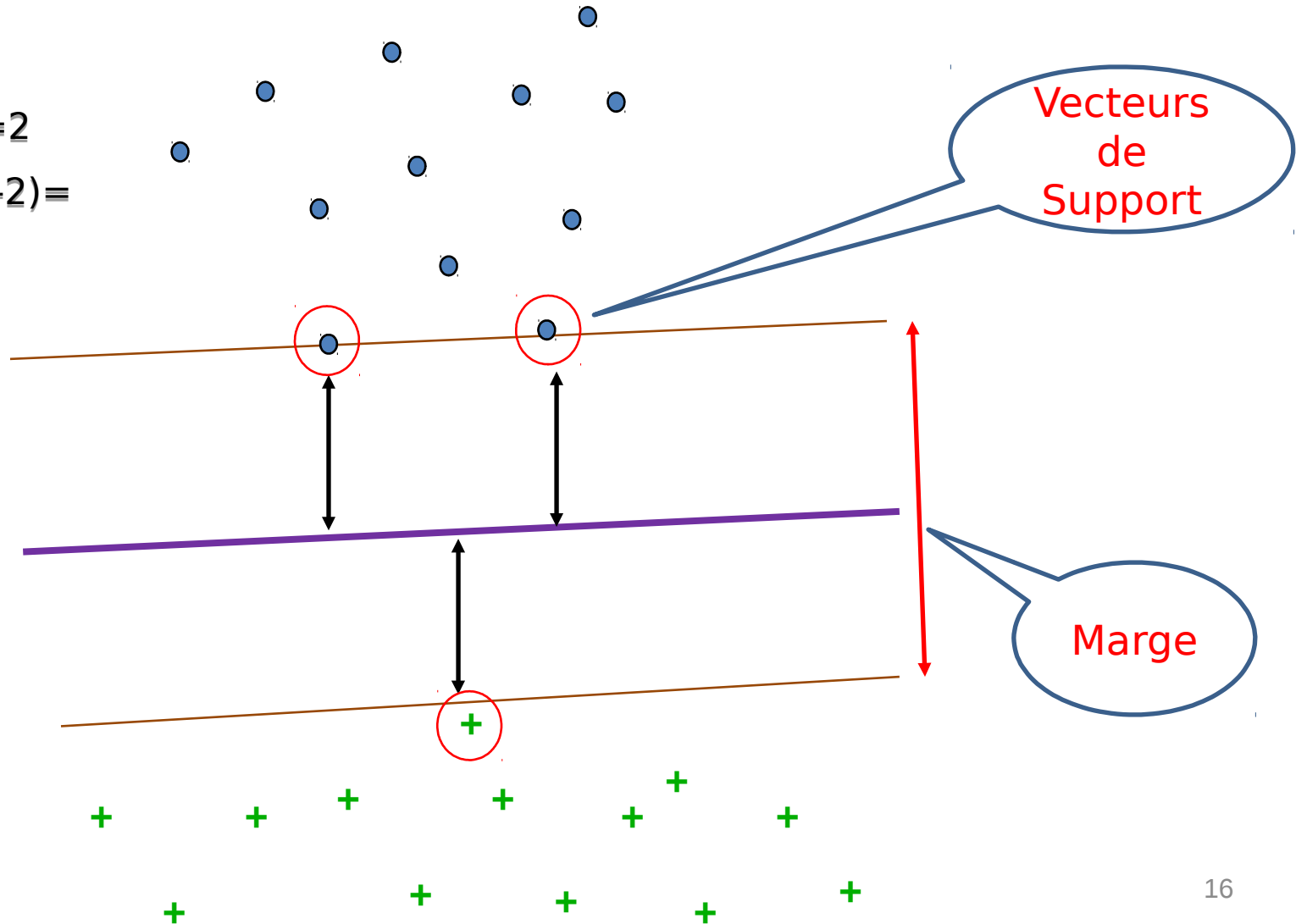
$$\Rightarrow w \cdot (x_1 - x_2) = 2$$

$$\Rightarrow w / \|w\| \cdot (x_1 - x_2) = 2 / \|w\|$$

$$wx + b = 1$$

$$wx + b = 0$$

$$wx + b = -1$$



SVM Linéaire (Sans erreurs)

- ✓ La valeur de la marge est inversement proportionnelle à $\|w\|$, maximiser la marge revient à minimiser $\|w\|^2$:

$$\begin{cases} \text{Min} & \frac{1}{2} \|w\|^2 \\ & y_i (\langle w, x_i \rangle + b) \geq 1, \quad \forall i \in [1..n] \end{cases} \quad (1)$$

Selon **Vapnik** :

Minimiser $\|w\| \Rightarrow$ Minimiser la + petite Dim-VC acceptable (de $f(x)$).

SVM Linéaire (Sans erreurs)

- ✓ On construit le Lagrangien L :

$$\left\{ \begin{array}{l} \text{Min } L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i (\langle w, x_i \rangle + b) - 1) \\ (y_i (\langle w, x_i \rangle + b) \geq 1) \\ \alpha_i \geq 0 \quad \forall i \in [1 \dots n] \end{array} \right. \quad (2)$$

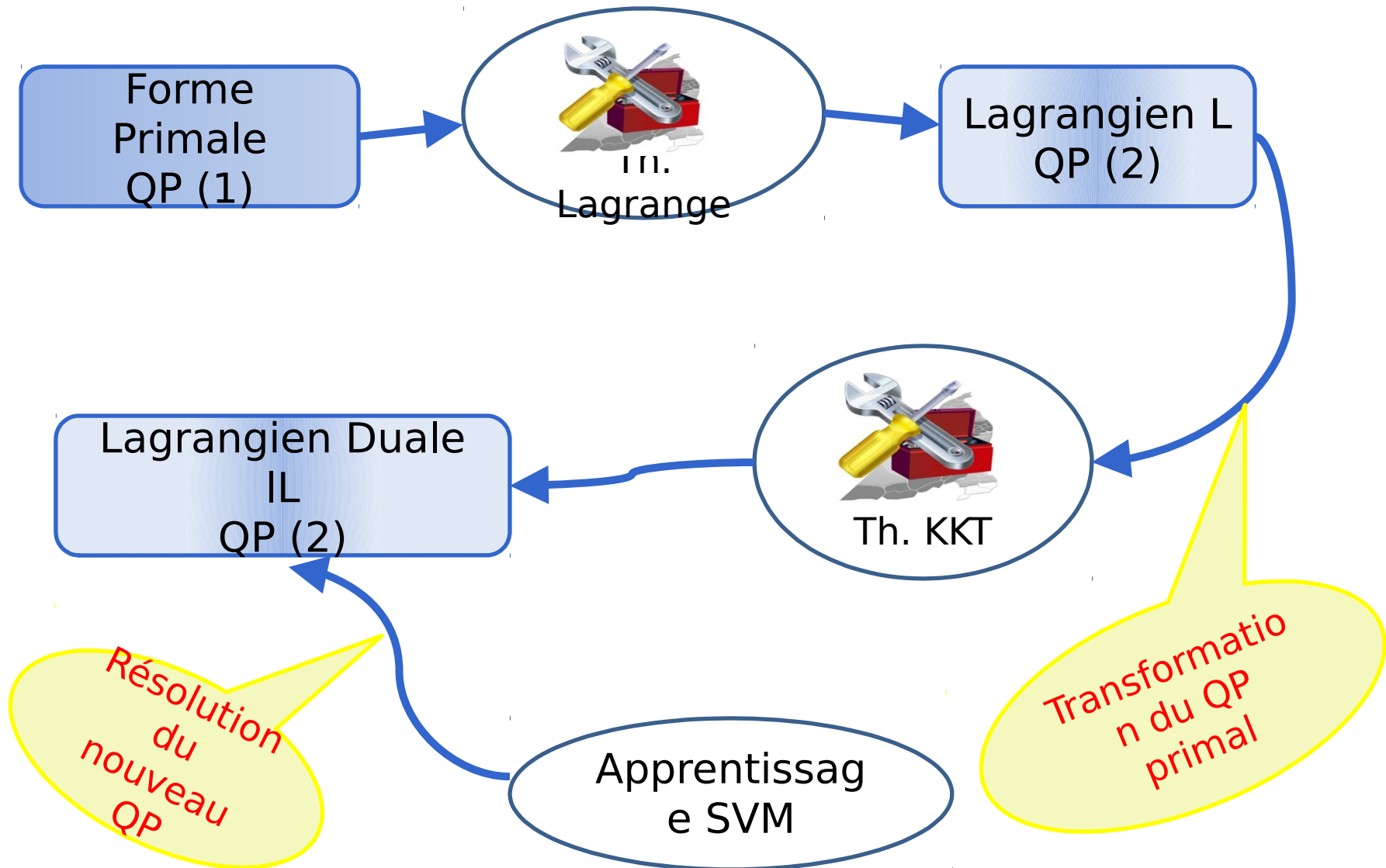
- ✓ D'après le Théorème de Karush-Kuhn-Tucker :

$$\left\{ \begin{array}{l} w^* = \sum_{i=1}^n \alpha_i y_i x_i \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{array} \right.$$

- ✓ En réinjectant ces valeurs dans (2), on obtient le lagrangien dual.

$$\left\{ \begin{array}{l} \text{Max } \mathcal{L}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\ \sum_{i=1}^n \alpha_i y_i = 0 \\ \alpha_i \geq 0, \quad \forall i \in [1 \dots n]. \end{array} \right. \quad (3)$$

SVM Linéaire (Sans erreurs)

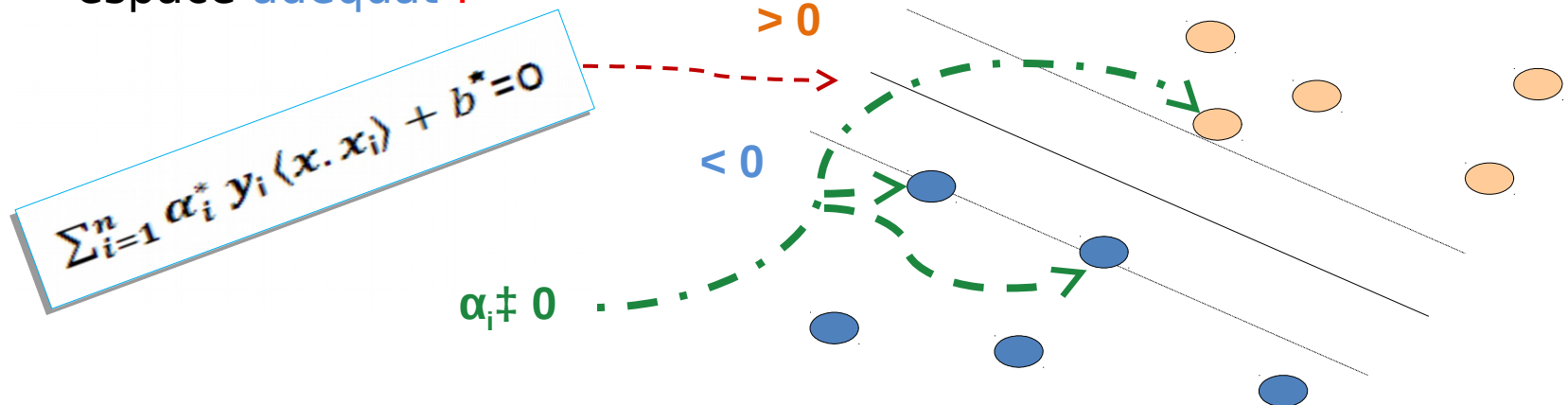


SVM Linéaire (Sans erreurs)

L'équation de l'hyperplan optimal :

$$\sum_{i=1}^n \alpha_i^* y_i \langle x, x_i \rangle + b^* = 0 \quad (5)$$

- ✓ Selon le **signe** de l'expression du membre gauche de l'équation, une **nouvelle** forme x présentée à la machine sera affectée au demi-espace **adéquat** !



- ✓ L'ensemble des points pour lesquels $\alpha_i \neq 0$, sont:
les « **points de support** » sur les bords de la marge.

SVM Linéaire (Avec erreurs)

- ✓ Relâcher les contraintes de marge en introduisant les variables ressorts:

$$\left\{ \begin{array}{l} \text{Min } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \\ y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i \\ C > 0, \xi_i \geq 0, \forall i \in [1..n] \end{array} \right. \quad (6)$$



- ✓ $\sum \xi_i$ constitue un coût caractérisant l'**erreur globale** commise sur A_n .
- ✓ C : paramètre de **régularisation**, borne les erreurs tolérées;
- ✓ Le QP est reformulé afin de **minimiser** une borne supérieure sur le R_{emp} , et **maximiser** la marge (tout en gardant petite la borne $>$ sur la dim-VC).

SVM Linéaire (Avec erreurs)

En suivant le même acheminement :



$$\left\{ \begin{array}{l} \text{Max } \text{IL}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle x_i \cdot x_j \rangle \\ \sum_{i,j=1}^n \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \quad \forall i \in [1..n] \end{array} \right. \quad (7)$$

La résolution du QP (7) permet de trouver la fonction d'**hyperplan optimal généralisé** en fonctions des α_i .

L'ensemble des points pour lesquels $\alpha_i \neq 0$, inclut :

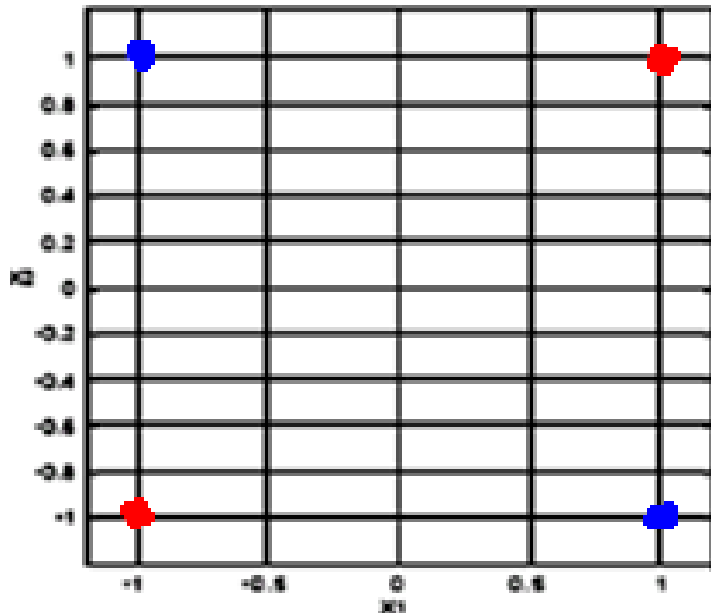
- ✓ les « **points de support** » habituels sur la marge;
- ✓ les exemples qui sont du **mauvais côté** de leur **marge** ($\alpha_i = C$).

SVM non Linéaire

✓ Pour les données **non** linéairement séparable:

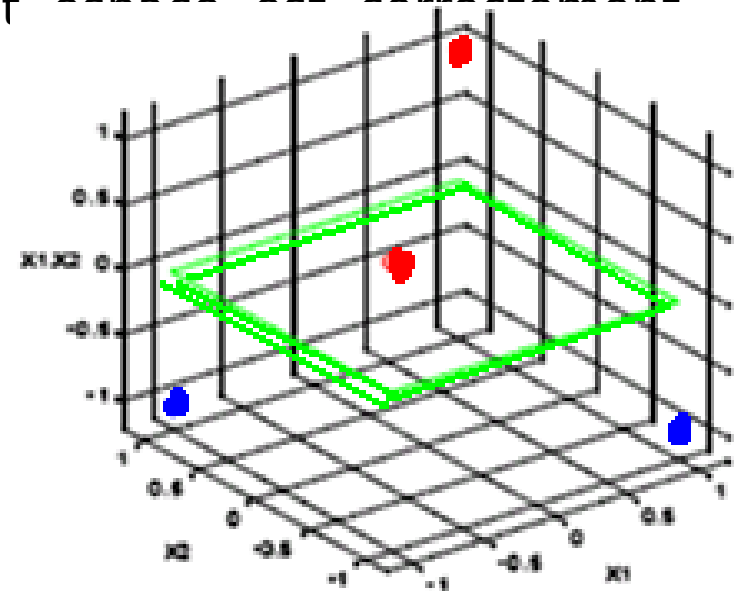
plonger les x_i dans un espace de représentation de dimension
>>

l'hyperplan optimal construit dans cet espace est correctement
généralisé



$\phi(x)$

A black arrow points from the 2D plot to the 3D plot, indicating the mapping function $\phi(x)$.



SVM non Linéaire

- ✓ En suivant le même acheminement:



$$\left\{ \begin{array}{l} \text{Max } \mathcal{L}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \phi(x_i) \cdot \phi(x_j) \\ \sum_{i,j=1}^n \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \quad \forall i \in [1..n]. \end{array} \right.$$

(8)

- ✓ Utilisation des noyaux avec des algorithmes d'apprentissage **linéaires** permet d'extraire des relations **non-linéaire**.

$$\left\{ \begin{array}{l} \text{Max } \mathcal{L}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \sum_{i,j=1}^n \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \quad \forall i \in [1..n]. \end{array} \right.$$

(9)

Conditions de Mercer

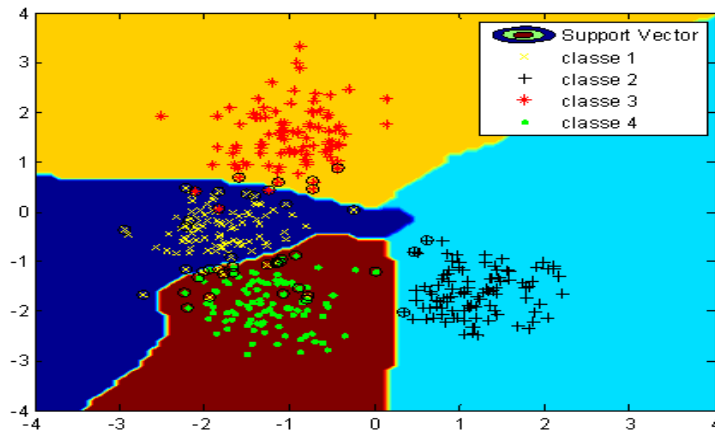
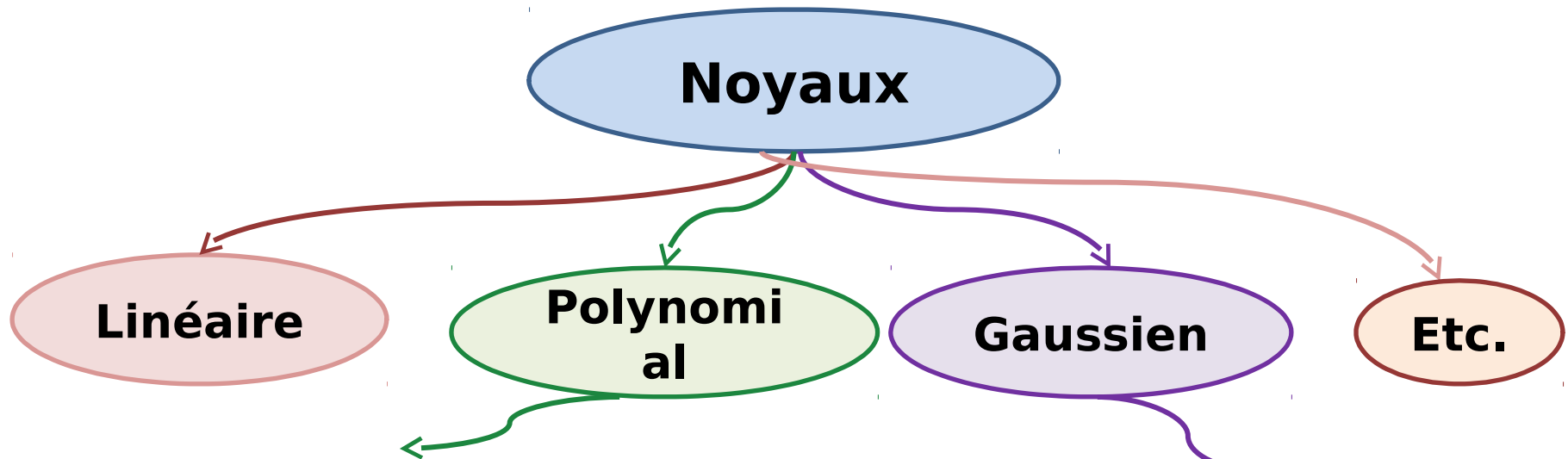
- ✓ Noyaux sont des fonctions de similarité respectant les **conditions de Mercer**

une fonction $K : X \times X \rightarrow \mathbb{R}$ est un noyau valide si elle est **symétrie** et **définie positive**.

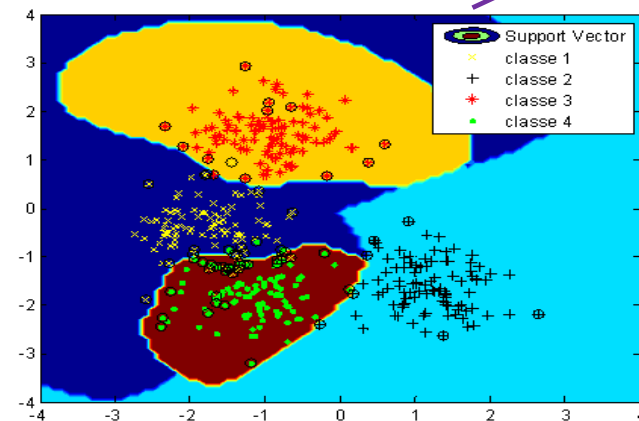
- ✓ Si l'ensemble d'apprentissage A_n est considéré :
la matrice dont les entrées représentent les similarités entre tous les objets, pris deux à deux, est appelée **matrice de Gram** des données ($G(i,j)=K(x_i,x_j)$, $i = 1$ à n):

$(G(i,j) = G(j,i) \text{ \& ttes les valeurs propres de « G » sont non nulles}).$

Noyaux (*Kernels*)

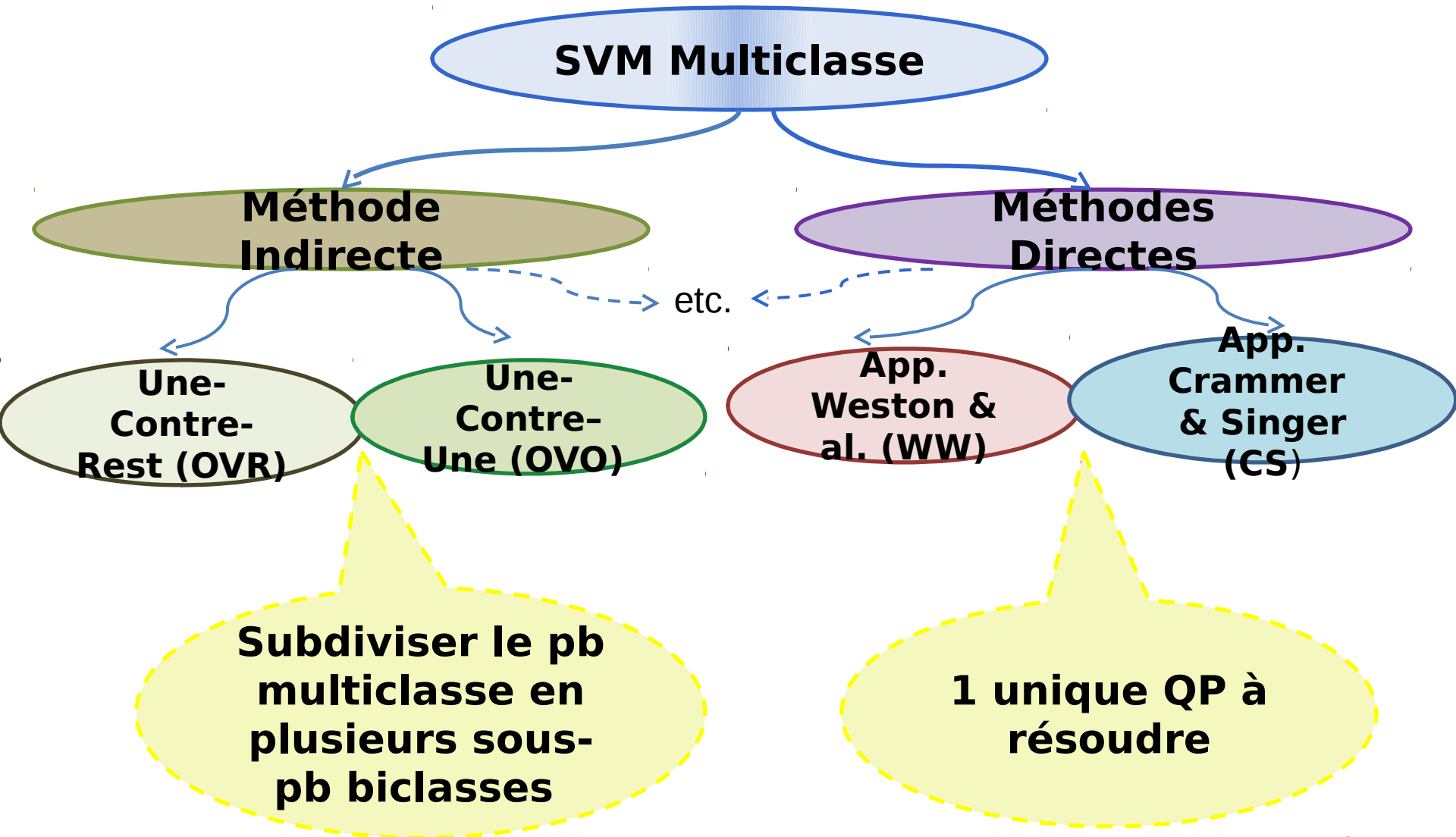


✓ C: 1000; kernel type:
poly
kernel option: 3.

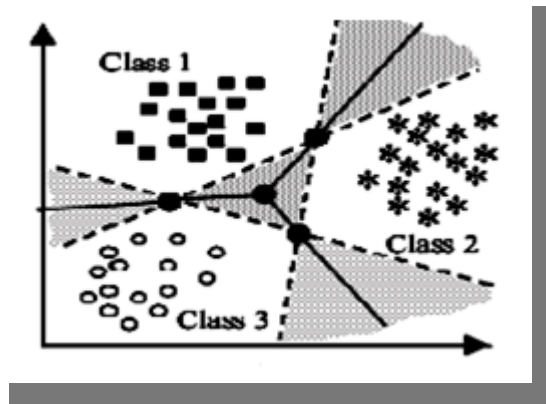


✓ C: 1000; kernel type:
gaussian
kernel option: 1.2.

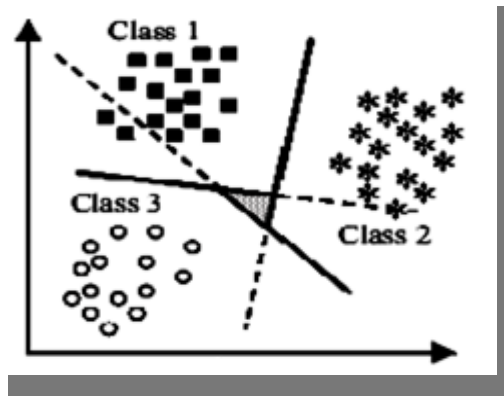
SVM Multiclasse



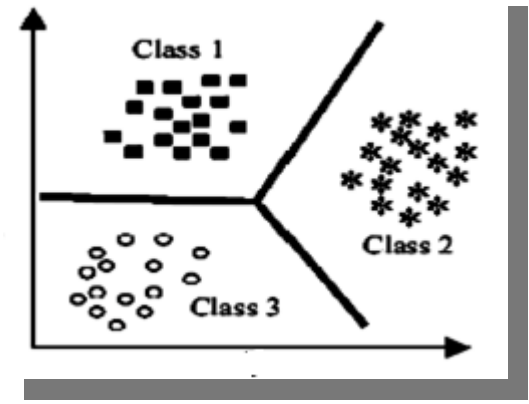
SVM Multiclasse



(a)



(b)



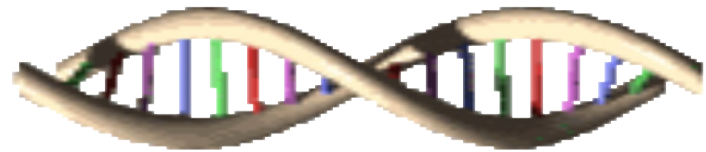
(c) (Statnikov, 2004)

- (a) La **SVM OVR** construit 3 classifieurs : classe 1 vs classes 2&3, classe 2 vs classes 1&3 et classe 3 vs classes 1&2.
- (b) La **SVM OVO** construit trois classifieurs : (classe 1 vs classe 2), (cl. 2 vs cl. 3) et (cl. 1 vs cl. 3).
- (c) La **SVM (WW) & (CS)** construit un classifieur unique en maximisant la marge simultanément entre toutes les classes.


SVM

- ✓ Pas de risque d'optimum local car optimisation d'un problème convexe;
- ✓ Justification théorique;
- ✓ Modularité : Séparateur + Optimisation + Noyau.

Algorithme Génétique (AG)



Algorithme génétique

- Introduit par **John Holland** professeur à l'université du Michigan, en 1975. 
- Son étudiant **De Jong K.A.** était le 1er à les utiliser pour **des problèmes d'optimisation**.
- Depuis l'apparition du livre de **Goldberg** en 1989 : «*Genetic Algorithms in Search, Optimization and Machine Learning*»,  ils sont devenus populaires.

Algorithme génétique

La mise en place d'un AG nécessite de répondre à ce genre de questions:

- Codage des solution ?
- Population initiale ?
- Fonction d'évaluation f ?
- Opérateurs génétiques ?
- Paramètres d'initialisation ?

On construit des chaînes (individu ou **chromosome**):

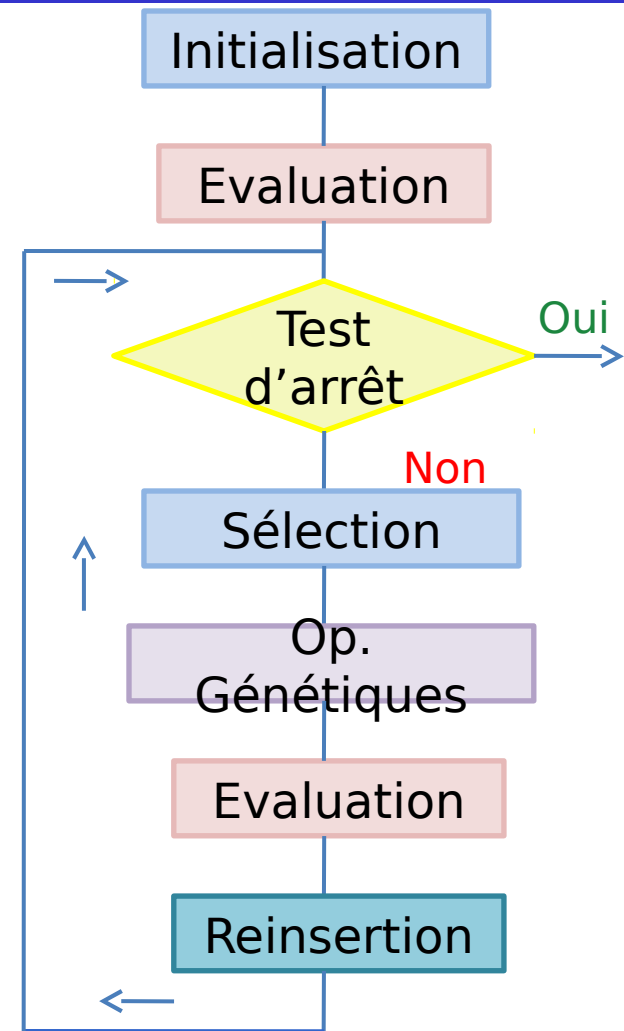
- **Binaires** (0,1);
- **Réelles.**

- **Solution :**



Algorithme génétique

- Après la génération d'une **population initiale**, on réalise une **évaluation** (f);
- On fait évoluer toute la population en même temps, par différents **opérateurs génétiques**;
- On remplace la population des parents en **réinsérant** les descendants;
- & l'algorithme reboucle.

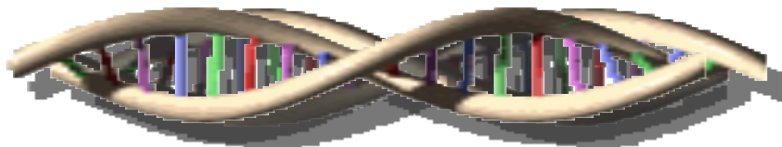


Théorème de *schemata*

Le **Théorème des *schemata***, met en évidence l'idée fondamentale de Holland ;

les **AGs** = une **compétition** entre deux phénomènes:

- la **destruction** de solutions par les opérateurs (mutation & croisement);
- et leur **maintien** par l'application de la sélection.



Théorème de

schemata

- Un **schéma** est un sous-ensemble de chromosomes qui partagent un ensemble particulier d'allèles définis sur $A_t = \{0,1,*\}$ où « * » représente « 1 » ou « 0 ».

✓ **Schéma** $S=00*010*1$:
 $\{00001001, 00101001, 00101011, 0001011\}$

✓ La **chaîne** $C= 00101001$ est 1 représentant du schéma S .

✓ L'**ordre** d'un schéma est le nbre de positions fixées (0,1)

$$O(00*010*1) = 6$$

✓ **Longueur** d'un schéma $\delta(S) =$ la distance entre la première & la dernière position fixée dans la chaîne:
 $\delta(00*010*1) = 8-1 = 7$;

Effet de la reproduction

- Reproduction : une chaîne est copiée avec $p_i = f_i / \sum f_i$
- Si la taille de la population $= n \Rightarrow$ on prédit que dans le temps $t+1$ (la génération suivante), le nombre des représentants est:

$$m(S, t+1) = m(S, t) \cdot n \cdot f(S) / \sum f_i \text{ \& on a: } f_{moy} = \sum f_i / n.$$

On aura : $m(S, t+1) = m(S, t) \cdot f(S) / f_{moy}$

- On considère l'hypothèse: $f(S) > c \cdot f_{moy}$ où c est une cste : $m(S, t+1) = m(S, t) \cdot (f_{moy} + c \cdot f_{moy}) / f_{moy} = (1+c) \cdot m(S, t)$

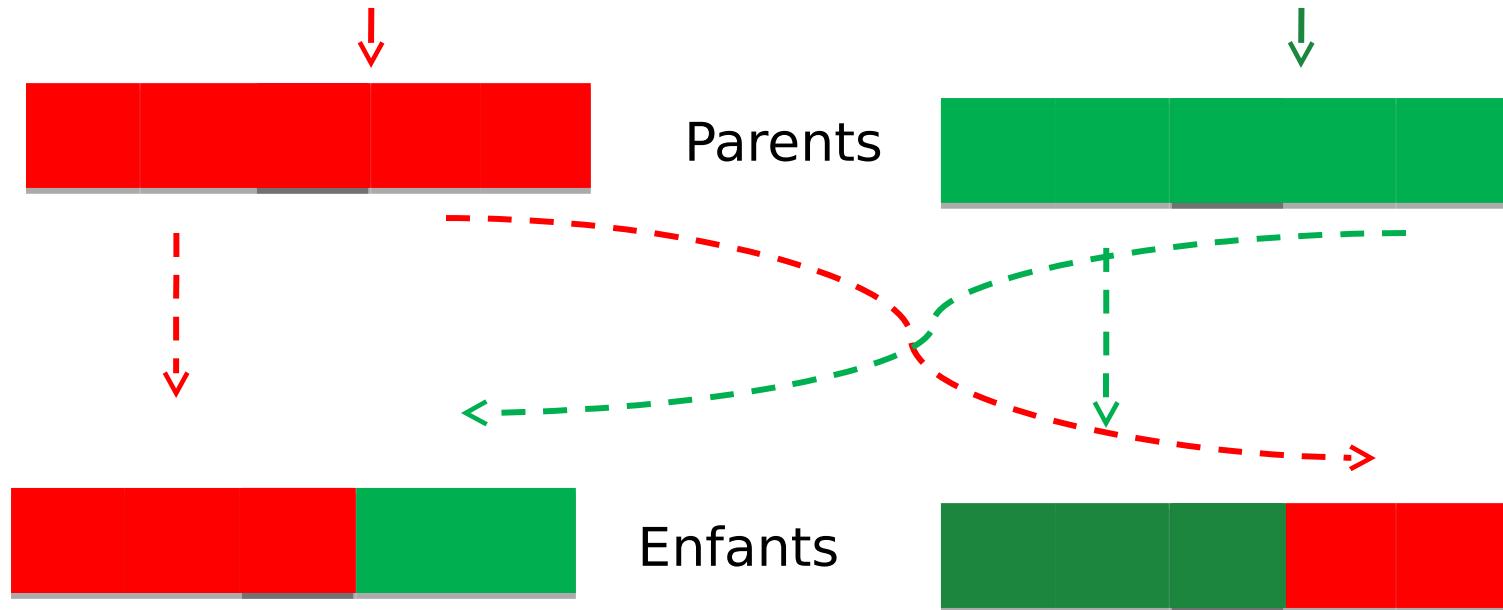
&

$$m(S, t) = m(S, 0) \cdot (1+c)^t$$

Effet de la reproduction

- ✓ Donc les schémas à valeur de fitness $>$ à la moyenne **augmentent** de manière **exponentielle** dans les nouvelles générations.
- ✓ (& ceux de valeur de fitness $<$ à la moyenne **disparaissent** de manière **exponentielle**).

Croisement



- Le **croisement** est un échange aléatoire structuré d'informations entre les chaînes.
- Il **crée** de nouvelles structures avec un minimum de perturbation.

Effet de Croisement

- C est représentant de S_1 & S_2
- Hypo: C participe à un croisement avec
- Si le pt de coupure intervient entre 4 & 5

C =	0	1	1	1	0	0	0	1
S_1 =	*	1	*	*	*	*	0	*
S_2 =	*	*	*	*	0	0	*	*

S_2 sera toujours représenté, par contre ce n'est pas obligatoire pr S_1 .

- On a 7 points de coupure possible: S_2 peut être détruit avec une prob. = 1/7 & S_1 détruit avec une prob. = 5/7;
- $\delta(S_2) = 1$ & $\delta(S_1) = 5$;
- Evaluation de la probabilité de survie/destruction d' un schéma:

$$P_{\text{destruction}}(S) = \delta(S)/(L-1) \Rightarrow (P_{\text{survie}}(S) = 1 - \delta(S)/(L-1))$$

- Si Prob.croisemt $P_{\text{croisemt}}: P_{\text{survie}}(S) > 1 - P_{\text{croisemt}}(S) \cdot \delta(S)/(L-1))$;

- $m(S,t+1) \geq m(S,t) \cdot (f(S)/f_{\text{max}}) \cdot (1 - (P_{\text{croisemt}}(S) \cdot \delta(S)/(L-1)))$

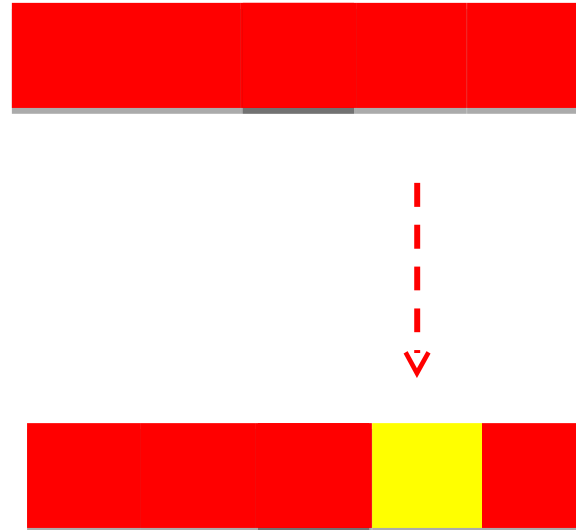
Effet de Croisement

- Plus la **fitness moyenne** d'un schéma est **élevée**,

plus sa longueur définie est **faible**

& **plus** ce schéma sera **représenté** dans la population suivante.

Mutation



- La **mutation** est l'altération aléatoire, à une probabilité p_{mutation} , d'une position singulière.

Effet de Mutation

- Un *schema* particulier survie lorsque chaque une des positions fixées $o(S)$ dedans survient aussi.
- En multipliant la probabilité de survie $(1 - p_{\text{mutation}})$ par elle-même $o(S)$ fois, on obtient:
la probabilité de la mutation de survie = $(1 - p_{\text{mutation}})^{o(S)}$.
- On conclut donc qu'un schéma particulier S reçoit un nombre prévu de copies dans la prochaine génération sous une reproduction (croisement & mutation) selon l'équation suivante :

$$m(S, t+1) \geq m(S, t) \cdot f(S) / f_{\text{moy}} \cdot (1 - (p_{\text{croisement}} \cdot \delta(S) / (L-1)) - (p_{\text{mutation}} \cdot o(S)))$$

Effet de Mutation

- Quelque soit le schéma court, à faible ordre, à haute moyenne reçoit des degrés croissants exponentiellement dans les générations suivantes.
- Cette conclusion est importante, à un point que ce théorème est considéré comme le:

« théorème fondamental de l'AG ».

Théorème de *schemata*

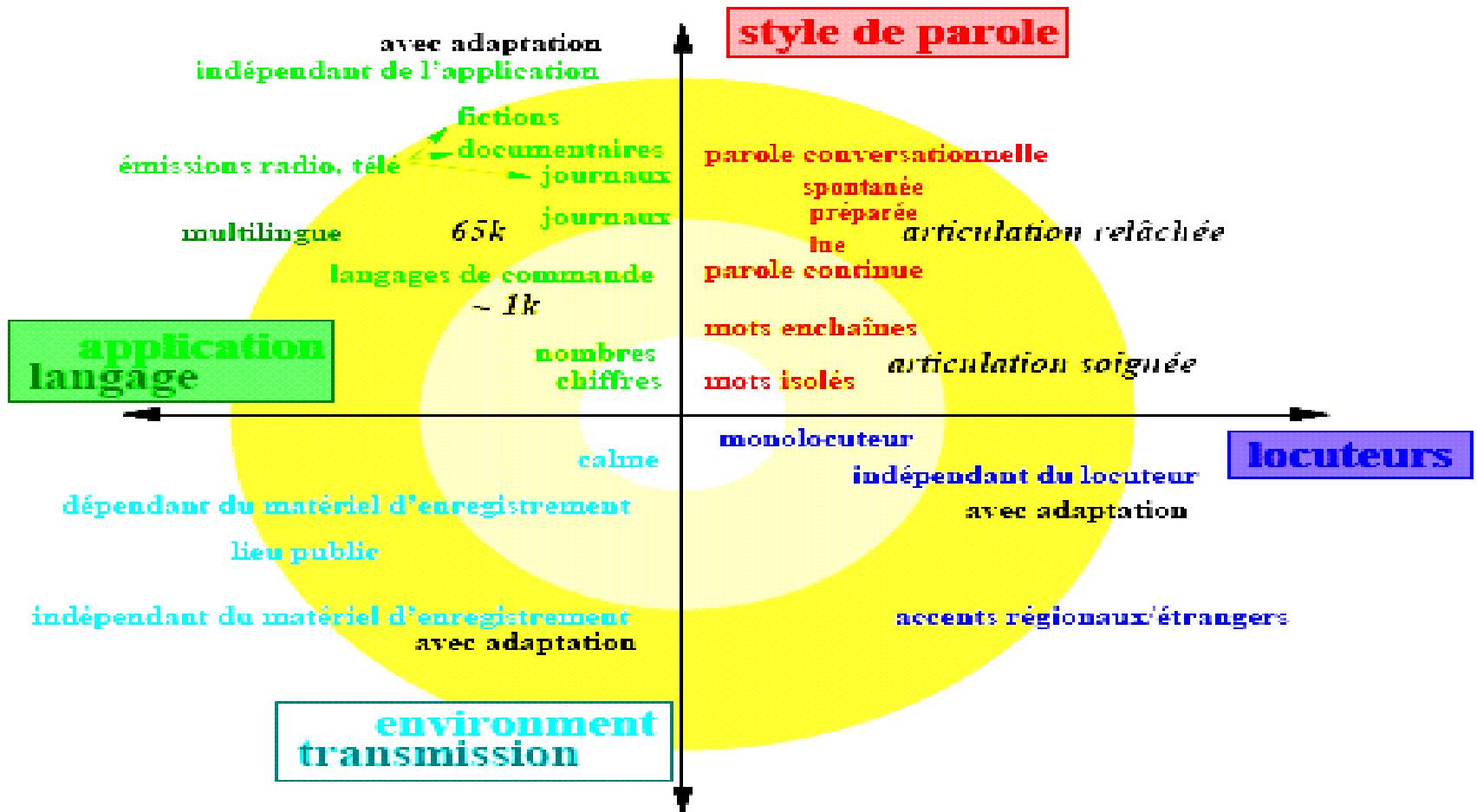
- Les AGs ne sont pas des boîtes noires. Ils s'appuient, sur une modélisation qui les rend beaucoup moins mystérieux.
- Ils sont des outils puissants aux fondements mathématiques.
- Des travaux plus récents basés sur :
 - ✓ Interprétation par Chaînes de Markov
 - ✓ Interprétation par énumération spatiale & transformée de Walsh.

Reconnaissance Automatique de la Parole (RAP)

Généralité sur la Parole

- **Parole** : est un moyen de communication simple & efficace;
- Elle est le **résultat** d'une action volontaire & coordonnée d'un certain nbre de muscles, sous contrôle du syst. nerveux;
- La **RAP** permet à la machine de reconnaître afin de traiter les commandes fournis par un (ou plusieurs) utilisateur humain dans différents scénarios.

Historique de la RAP



1^{er} sys. RAP : (1952) Davis et al. aux Bell Labs -> (0-9) pour 1 seule voix

Application de la RAP Audio

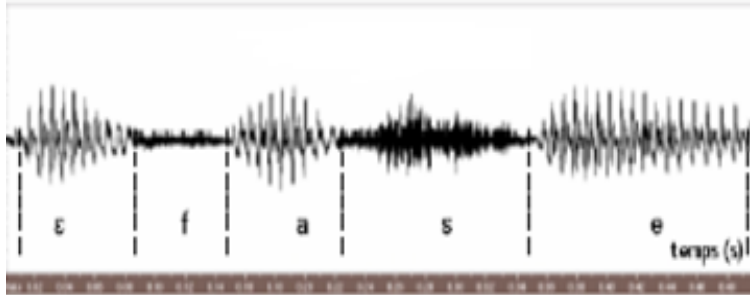
- **Reservation** de billets d'avion, de train, d'hôtel;
- **Consultation** et transaction bancaire & boursières;
- Commande à distance d'automate (**Industrie pétrolière**);
- **Contôle d'avion** (avec une libération visuelle du pilote);



trangère (bonne

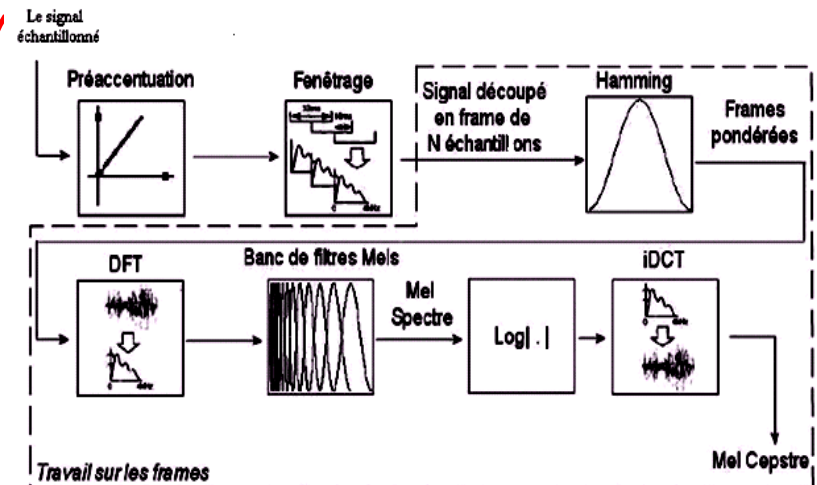


Signal de parole



Traitement
Automatique
(**Acoustico-**
phonétique)

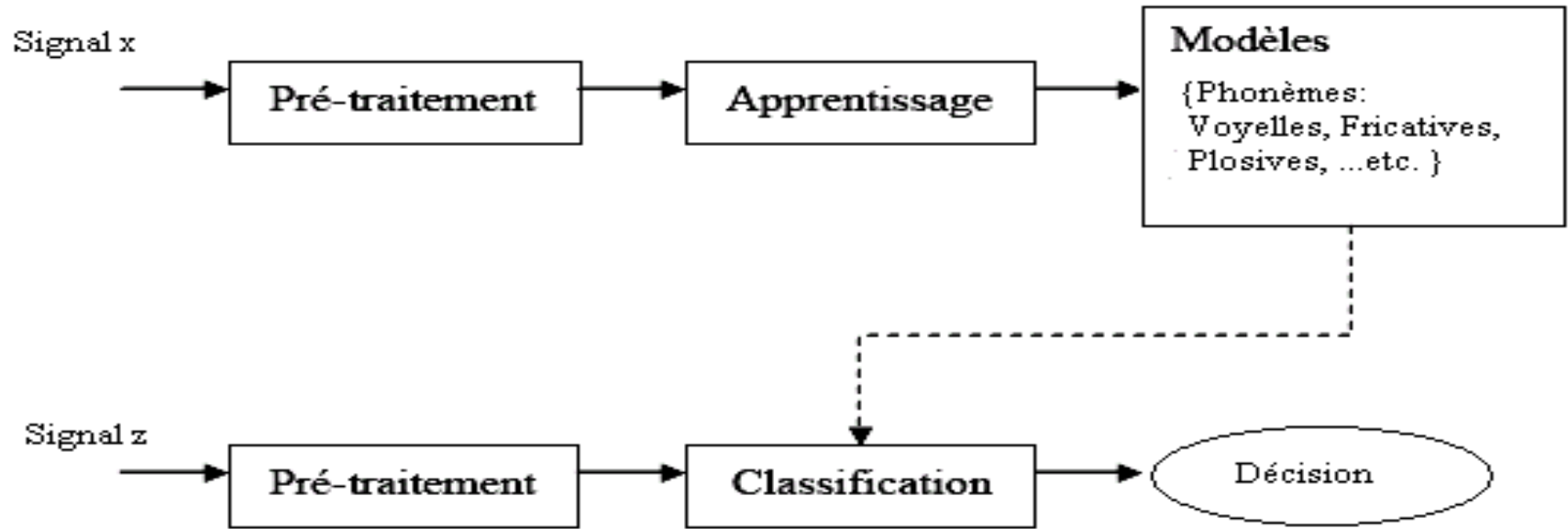
Reconnaissance
Automatique
de la
parole



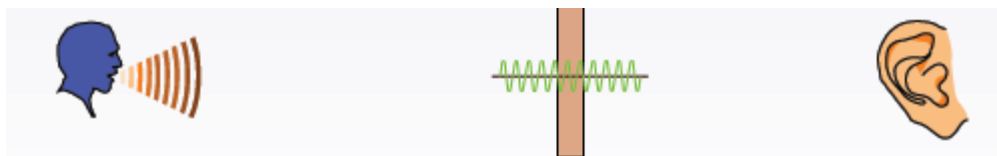
Extraction des **MFCC**

- ✓ **Segmenter** le signal en **unités linguistiques** (syllabe, phonème, phone);
- ✓ Alphabet International (**API**) = {Voyelles, consonnes, plosives, fricatives, etc}.

Reconnaissance Automatique de la Parole



- ✓ L'information contenue dans le vecteur acoustique est évaluée, pour prendre une décision sur la classe à laquelle il appartient.



Reconnaissance Automatique de la Parole

- ✓ Les techniques utilisés pour la RAP:
 - Approche **vectorielle** (DTW, VQ);
 - Approche **connexionniste** (RN temporel: TDNN, TDRBF);
 - Approche **stochastique**(HMM, RB, GMM);
 - Méthodes à base de noyaux (**SVM**)



Systemes d'optimisation proposés


(Sélection d'instance

Sélection de variables)

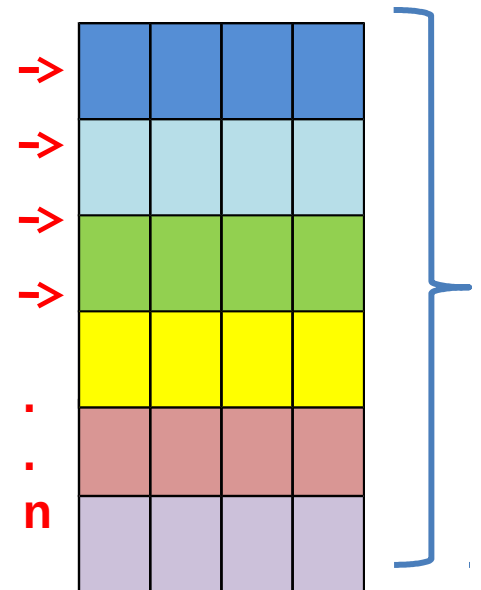
Système d'optimisation (GA-SVM-IS)

- **Hybridation** entre **SVM** & **AG** :



Sélection des **instances** (*instance selection*)

Technique permettant de choisir les instances les plus **intéressantes** afin de faciliter la classification.



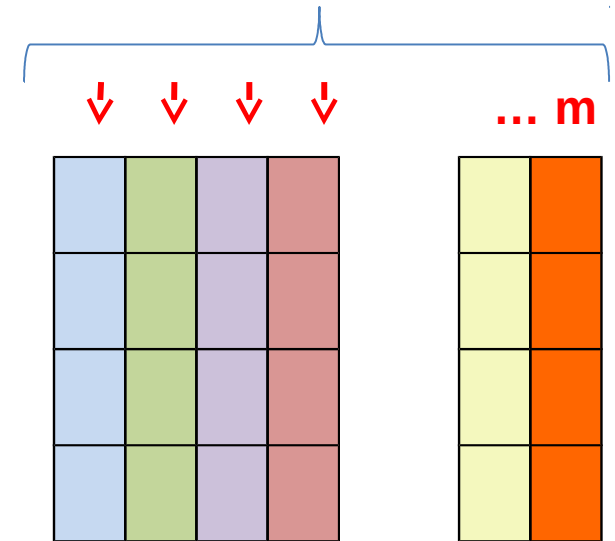
Système d'optimisation (GA-SVM-FS)

- **Hybridation** entre **SVM** & **AG** :



Sélection de Variables (*feature selection*)

Technique permettant de choisir les variables les plus **informatives** et éliminer les **redondantes** & indépendantes de la classe.



Systemes d'optimisation

- ✓ La **différence technique** entre les deux systèmes: la **longueur des chromosomes** qui sont utilisés pour la construction des sous-ensembles d'apprentissage.

GA-SVM-IS



Chromosome ->
chaîne binaire (0,1) à longueur
=**n**;

n : nombre d'**instances** dans
l'ensemble d'apprentissage.

GA-SVM-FS

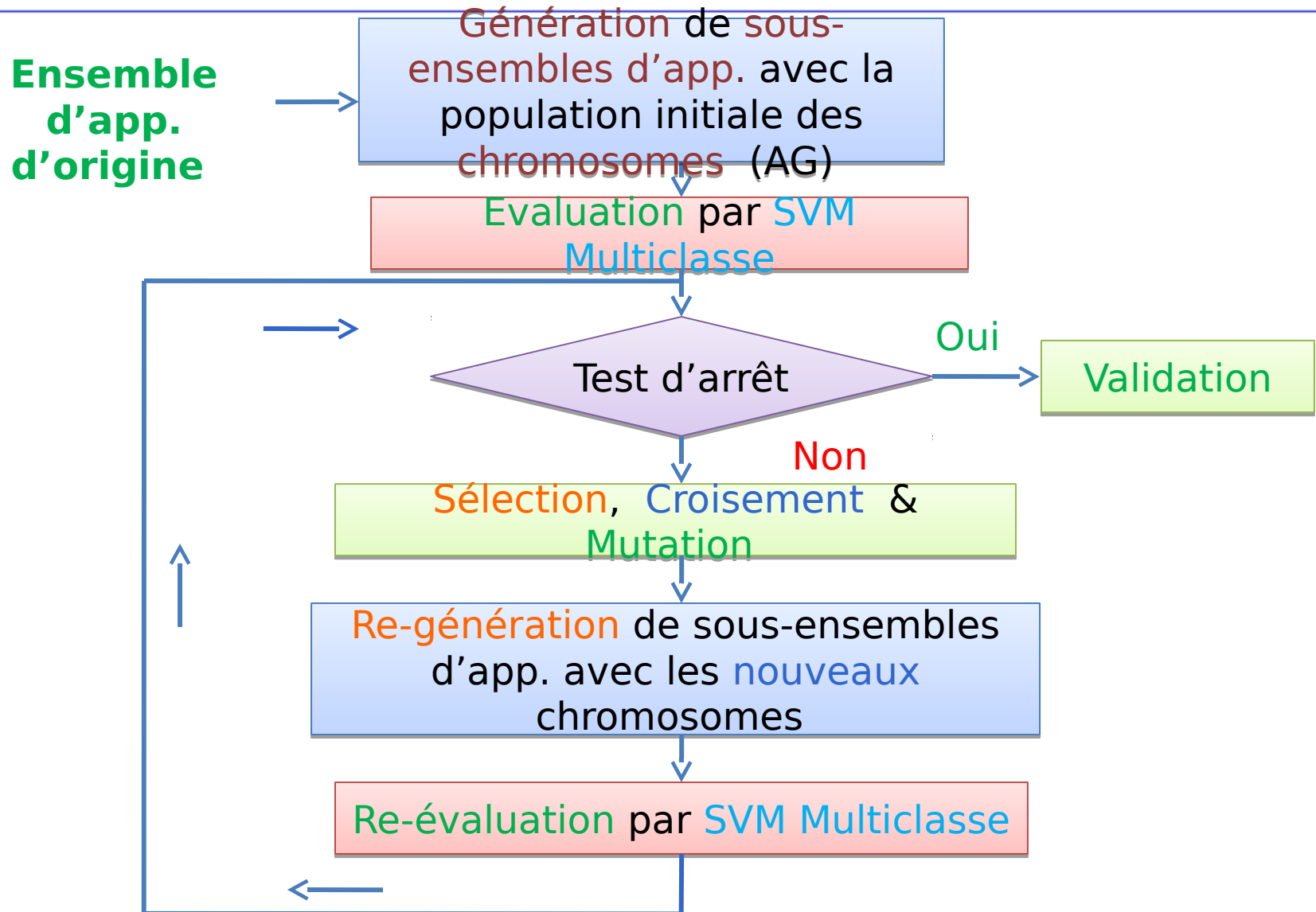


Chromosome ->
chaîne binaire à longueur =**m**;

m : nombre de **variables** dans
l'ensemble d'apprentissage.

Pour RAP: m nombre de **blocs MFCC**.

Systemes d'optimisation

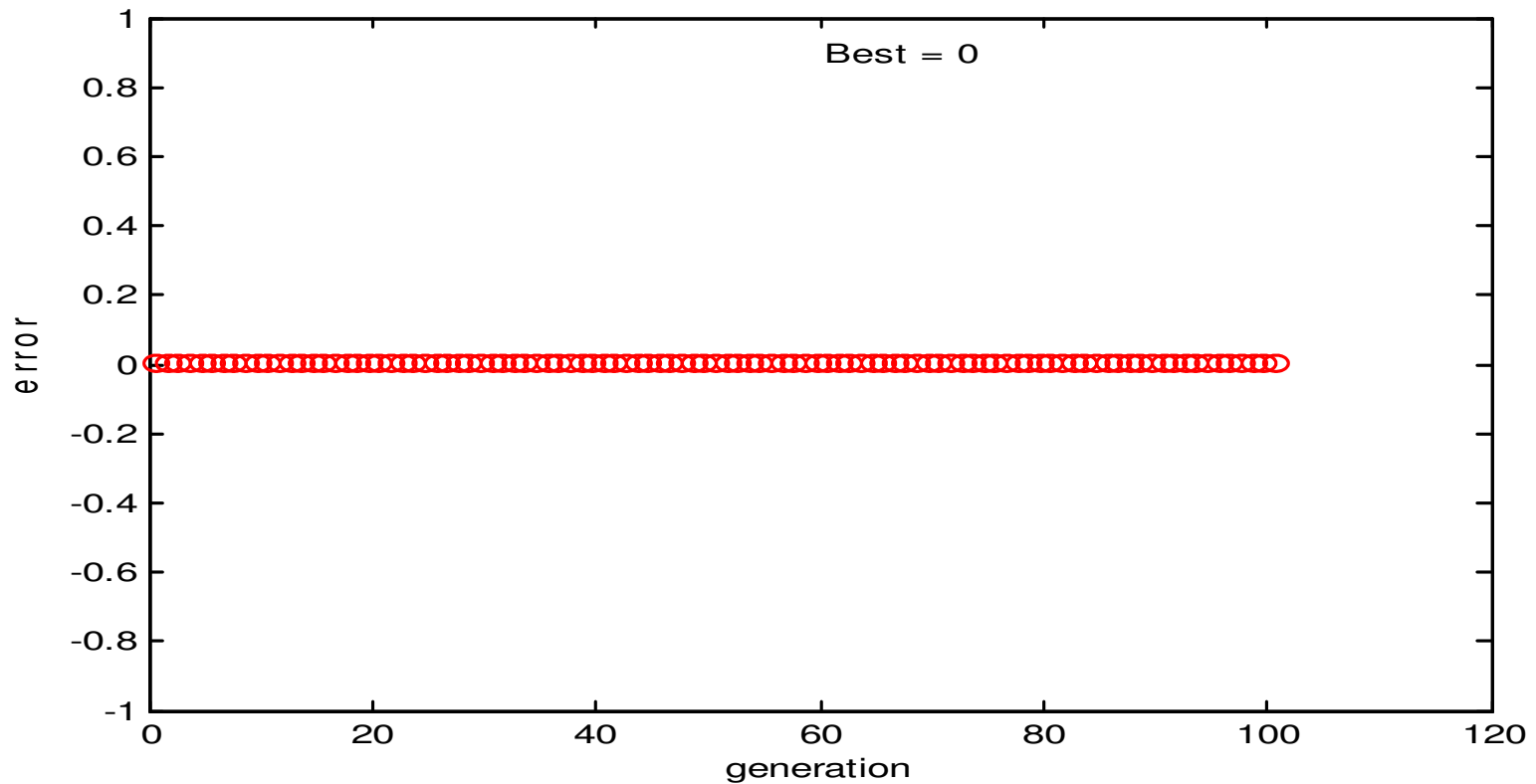


Expériences & Résultats

IRIS Plants

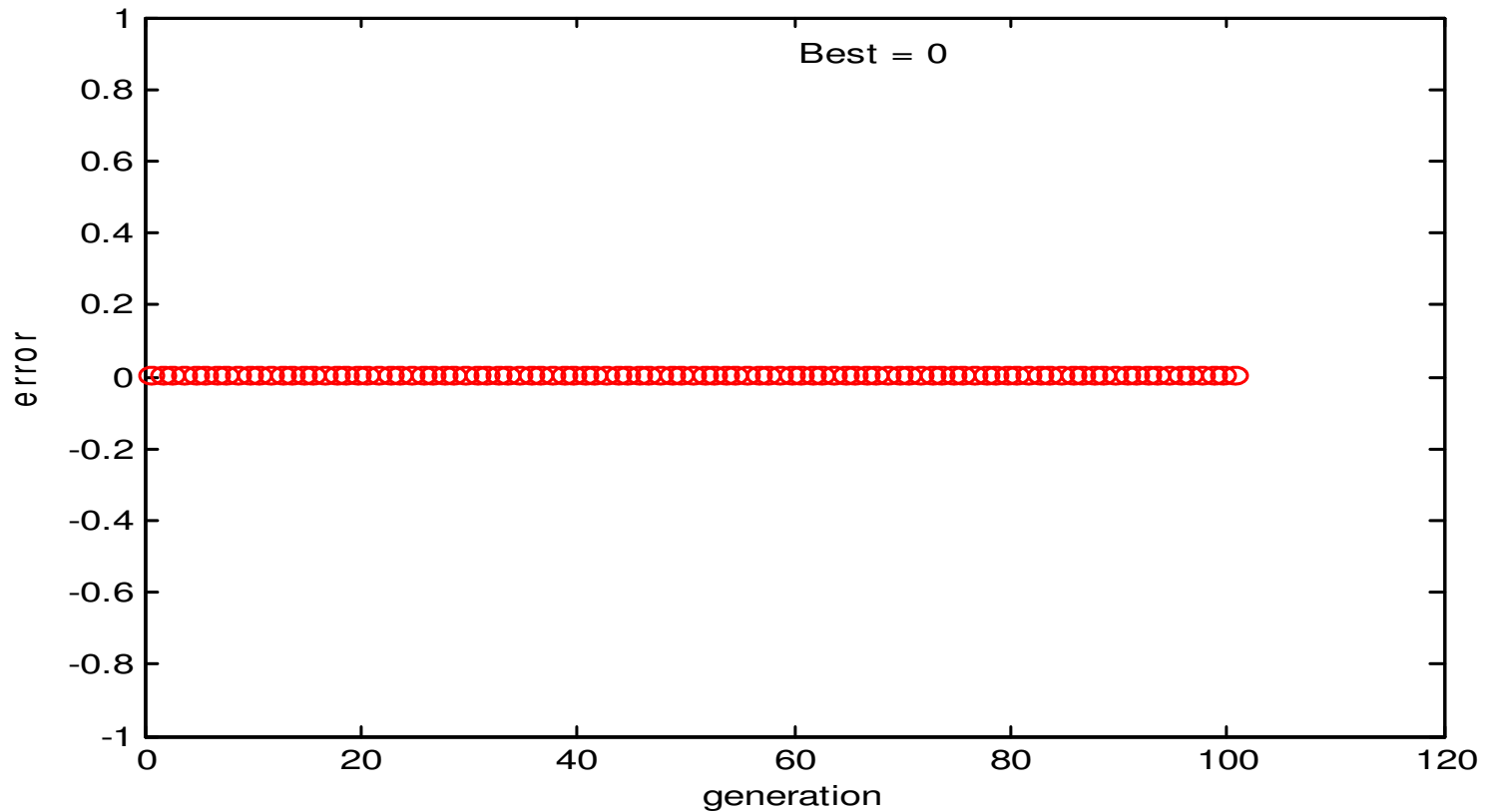
- Base de données Botanique: ***Iris Plants (150, 4)*** ;
- 3 classes -> type de plante d'iris :
 « **Setosa** », « **Versicolour** » et
 « **Virginica** »;
- 50 **instances** pour chaque classe (App.: 35 & Test: 15);
- Chaque instance est composés de 4 **variables** (*features*) numériques en cm :
 longueur de la sépale, largeur de la
 sépale,
 longueur de la pétale, et longueur de la

Résultats sur DB: *IRIS Plants*



Taux d'erreur de l'hybridation GA-SVM-IS pour Iris Data.

Résultats sur DB: *IRIS Plants*



Taux d'erreur de l'hybridation GA-SVM-FS pour Iris Data.

Résultats sur DB: *IRIS Plants*

Système Paramètres	SVM Multiclass : One-vs.-One (KM-SVM)	GA-SVM-IS	GA-SVM-FS
Taux de reconnaissance (%)	100 100 93.33	100 100 100	100 100 100
	97.77	100	100
Temps d'apprentissage (sec.)	0.26	0.02	0.15
Temps de test (sec.)	0.05	0.001	0.01
Taille de l'ensemble d'apprentissage	Ensemble d'orgine (105, 4)	Sous-ensemble réduit1 (43,4)	Sous-ensemble réduit 2 (105,1) [0 0 0 1]

Corpus TIMIT

Un corpus de 18 phonèmes (6 voyelles, 6 fricatives et 6 plosives) de la base de données américaine TIMIT.

- ✓ « **Corpus TIMIT (a)** à 13 composants» :
12 MFCC + énergie.
- ✓ « **Corpus TIMIT (b)** à 39 composants» :
12 MFCC + énergie + $\partial 1$ + $\partial 2$

Corpus TIMIT

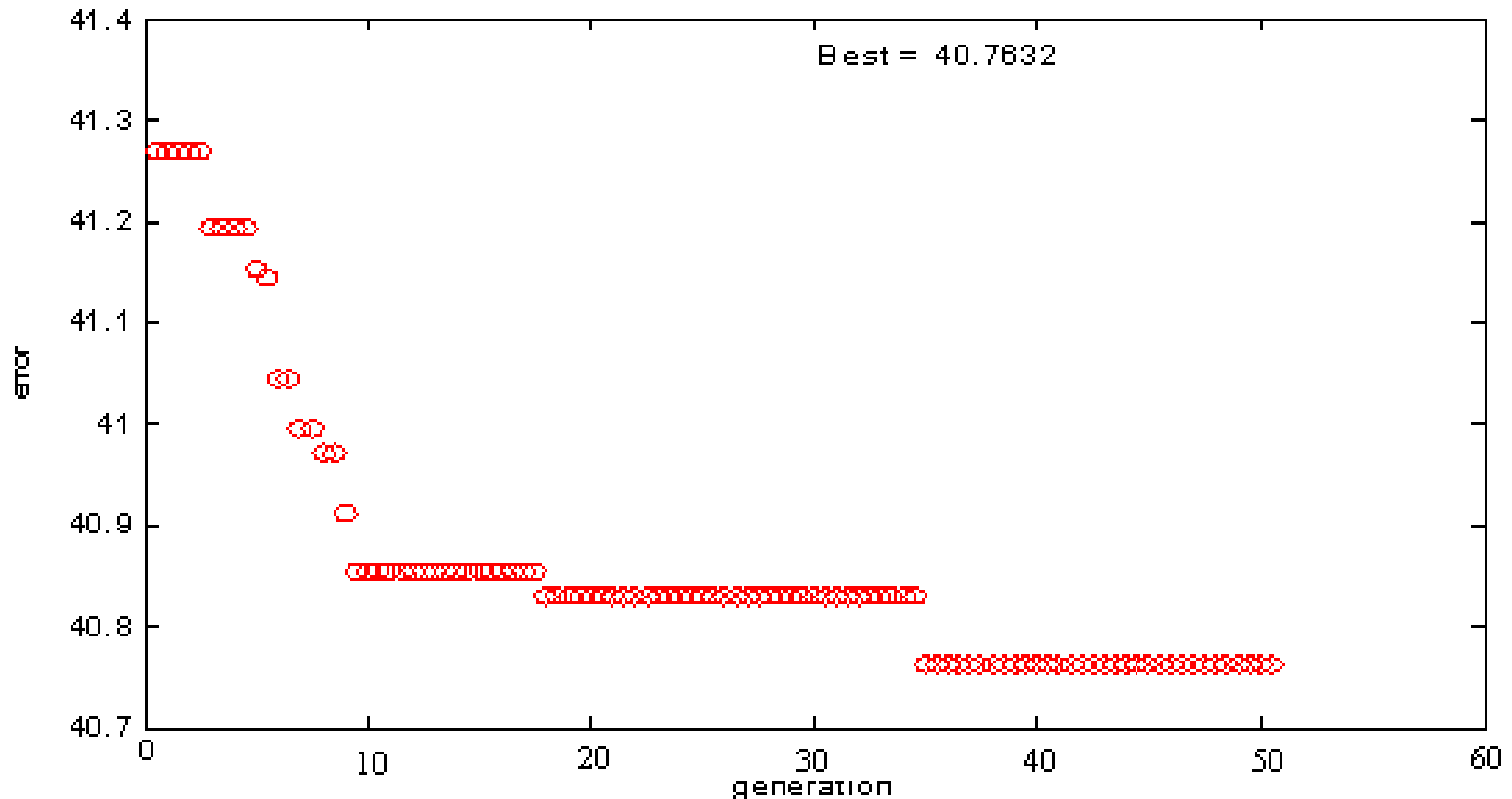
Classe	Phonème	Train	Test
Voyelles	/ah/	2200	879
	/aw/	700	216
	/ax/	3352	1323
	/ax-h/	281	95
	/uh/	502	221
	/urw/	536	170
Fricatives	/dh/	2058	822
	/f/	2093	911
	/sh/	2144	796
	/v/	1872	707
	/z/	3574	1273
	/zh/	146	74
Plosives	/b/	399	182
	/d/	1371	526
	/g/	1337	546
	/p/	2056	779
	/q/	3307	1191
	/t/	3586	1344
Totaux d'occurrences		31514	12055

Corpus TIMIT (a)

Classe	Phonème	Train	Test
Voyelles	/ah/	2195	879
	/aw/	699	216
	/ax/	3446	1346
	/ax-h/	361	118
	/uh/	502	221
	/urw/	536	170
Fricatives	/dh/	2688	1053
	/f/	2093	912
	/sh/	2138	796
	/v/	1883	710
	/z/	3571	1273
	/zh/	146	74
Plosives	/b/	2067	886
	/d/	3376	1245
	/g/	1929	755
	/p/	2475	957
	/q/	3435	1244
	/t/	4154	1535
Totaux d'occurrences		37694	14390

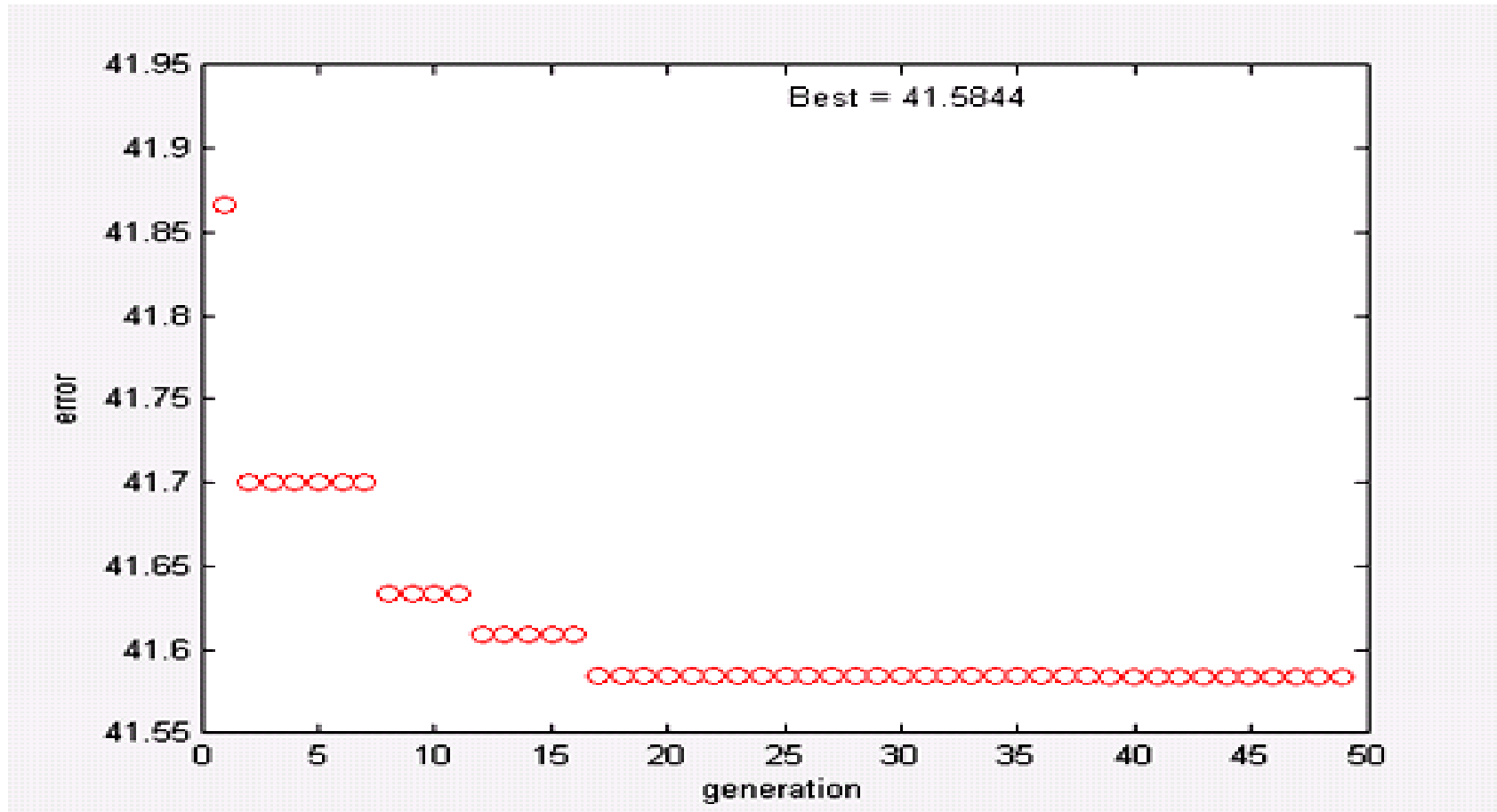
Corpus TIMIT (b)

Résultats sur « **Corpus TIMIT (a)** à 13 MFCC »



Taux d'erreur de l'hybridation **GA-SVM-IS** pour le corpus (a) global

Résultats sur « **Corpus TIMIT (a)** à 13 MFCC »

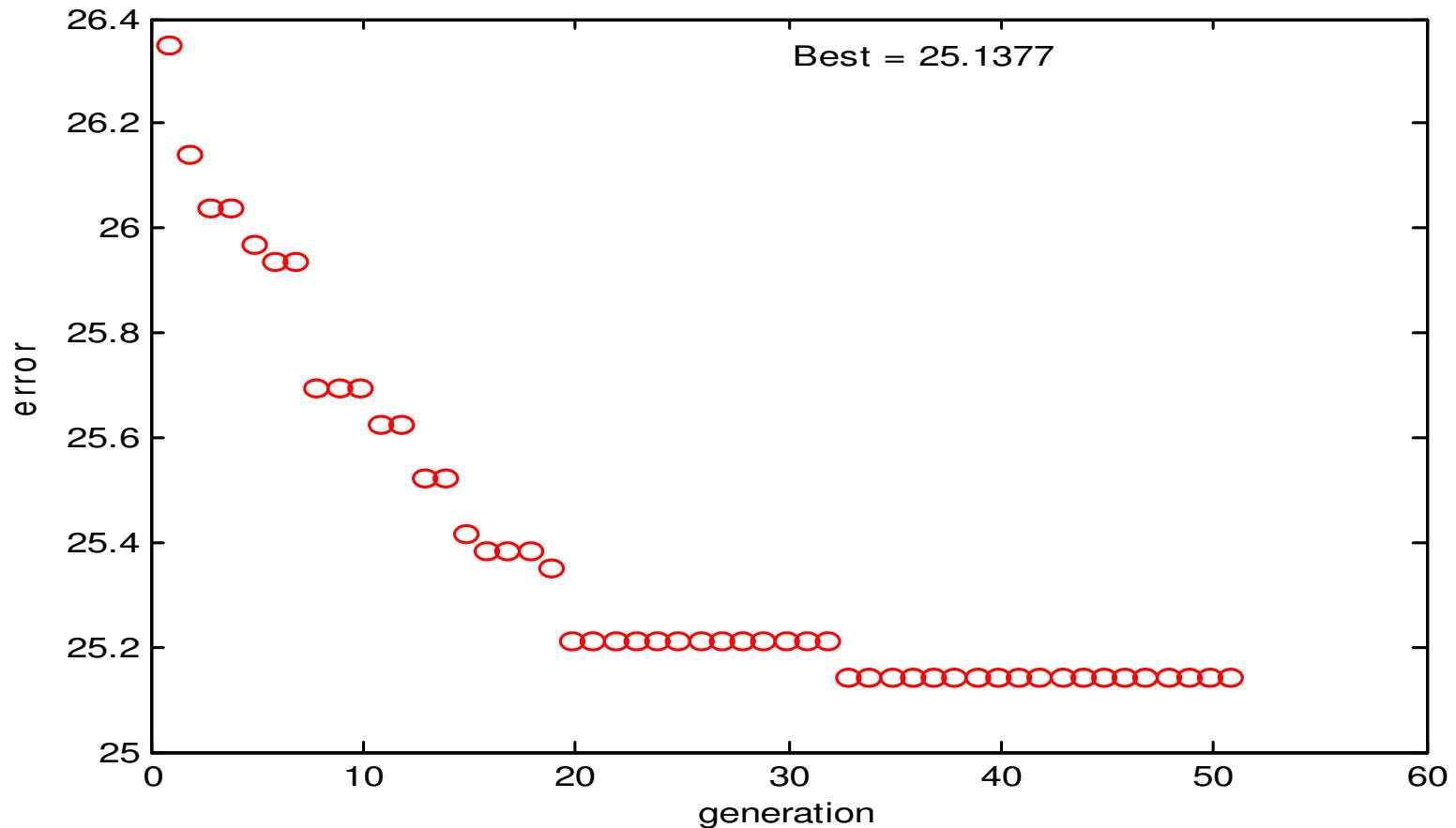


Taux d'erreur de l'hybridation **GA-SVM-FS** pour le corpus (a) global

Résultats sur « **Corpus TIMIT (a)** à 13 MFCC »

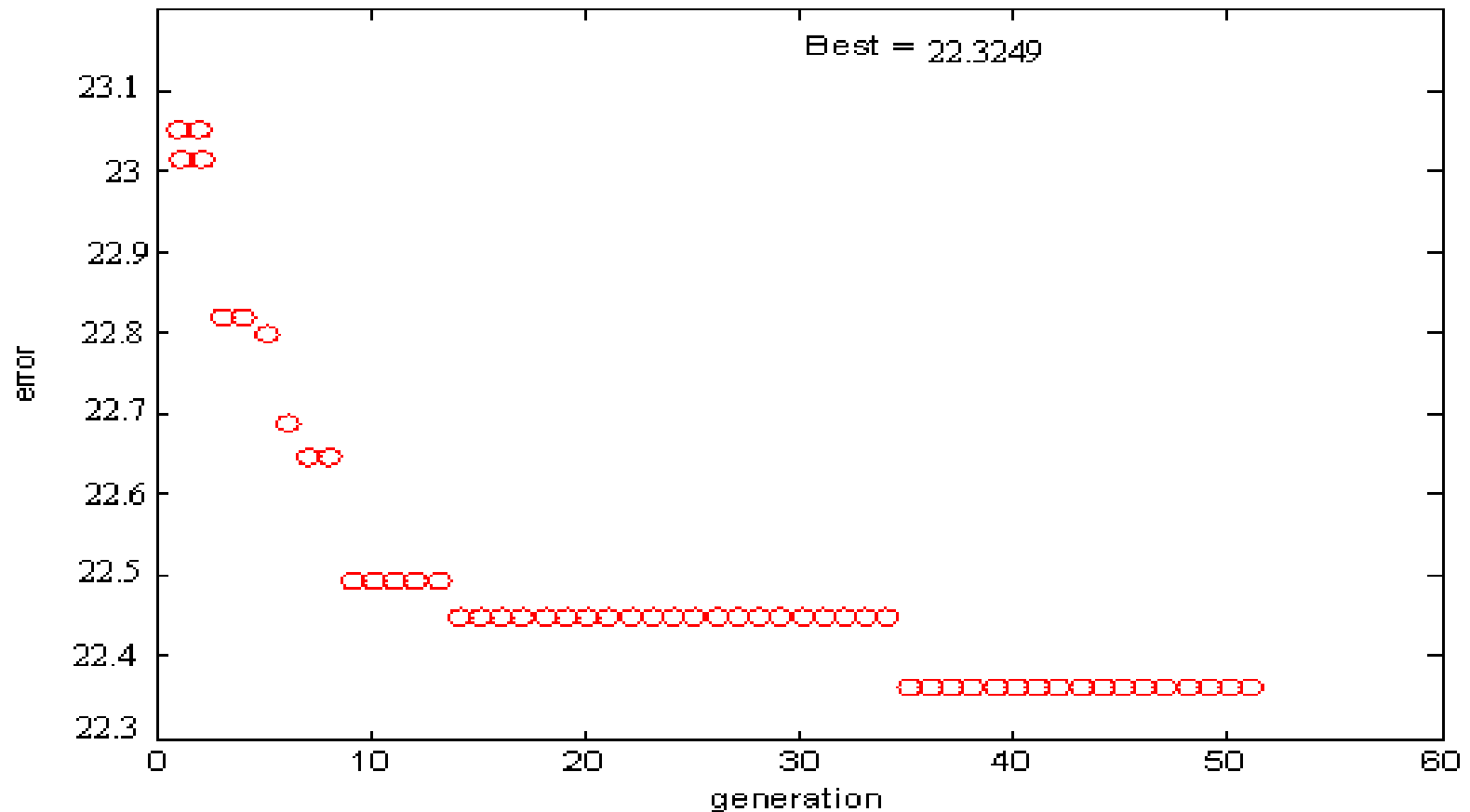
<div>Paramètres</div> <div>Système</div>	Taux de reconnaissance (%)	Temps Train (Sec.)	Temps Test (Sec.)	Complexité du modèle		Taille de l'ensemble d'apprentissage
				# VS	Taille de Matrice GRAM	
SVM (CS- Light Joachim)	58.30	66.55	0.19	42	31514 x 31514	Ens. App. d'origine (31514, 416)
GA-SVM-IS	59.24	37.20	0.22	40	15684 x15684	Sous-ens1. (15684 , 416)
GA-SVM-FS	58.42	55.83	0.22	36	31514 x 31514	Sous-ens2. (31514, 208)

Résultats sur « **Corpus TIMIT (a)** à 13 MFCC »



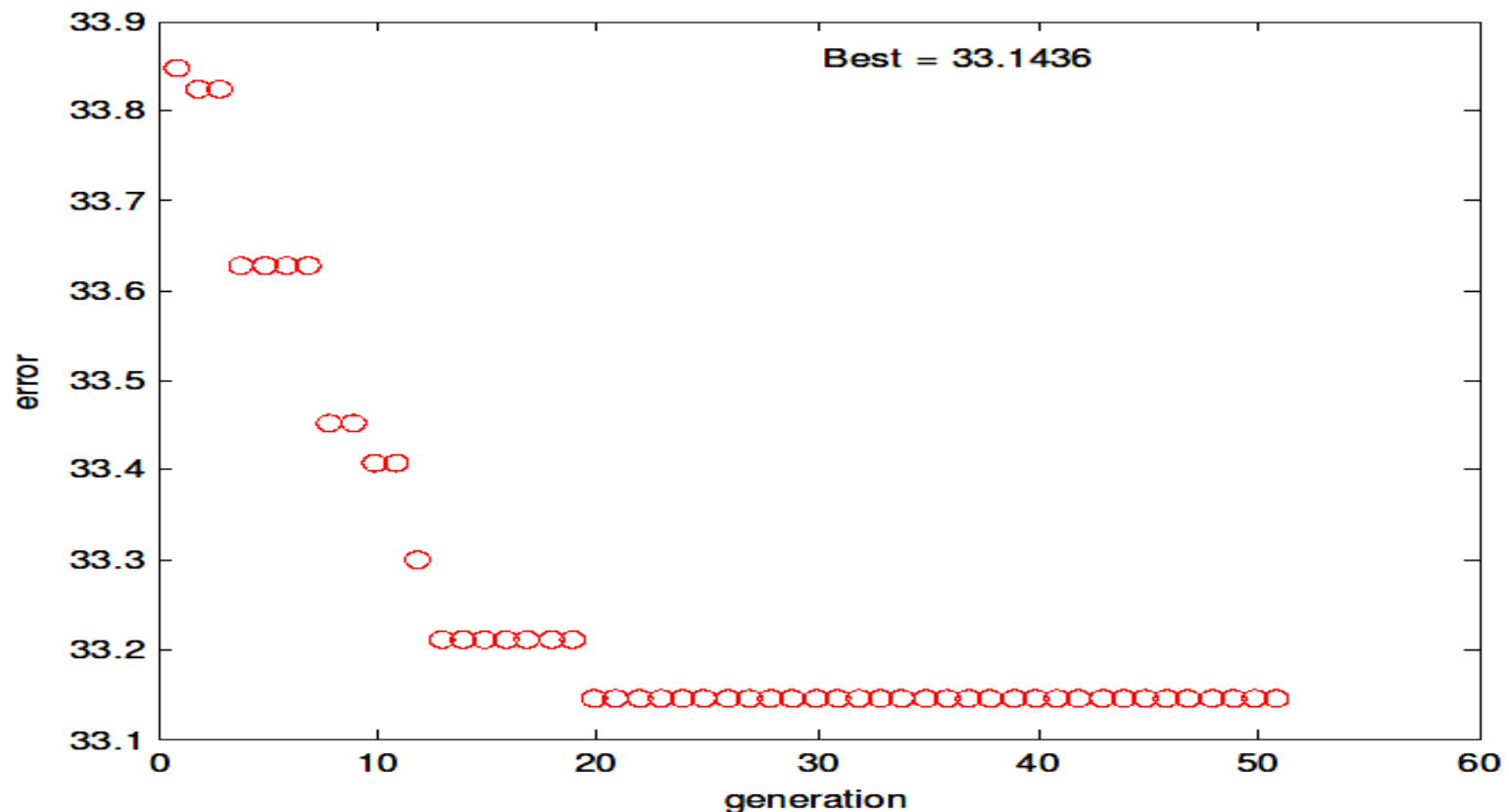
Taux d'erreur de l'hybridation **GA-SVM-IS** pour les **voyelles**

Résultats sur « **Corpus TIMIT (a)** à 13 MFCC »



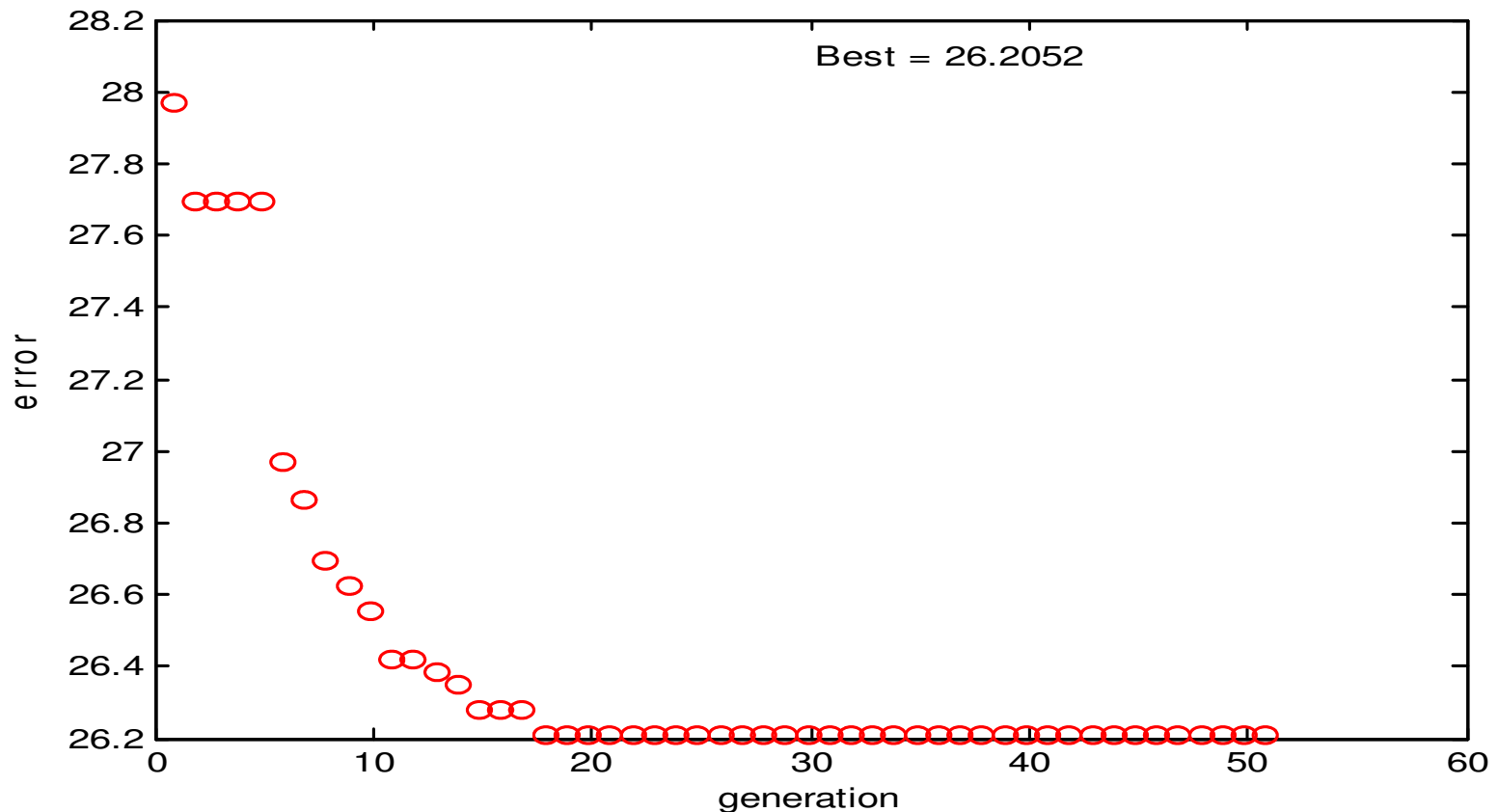
Taux d'erreur de l'hybridation GA-SVM-IS pour les **fricatives**

Résultats sur « **Corpus TIMIT (a)** à 13 MFCC »



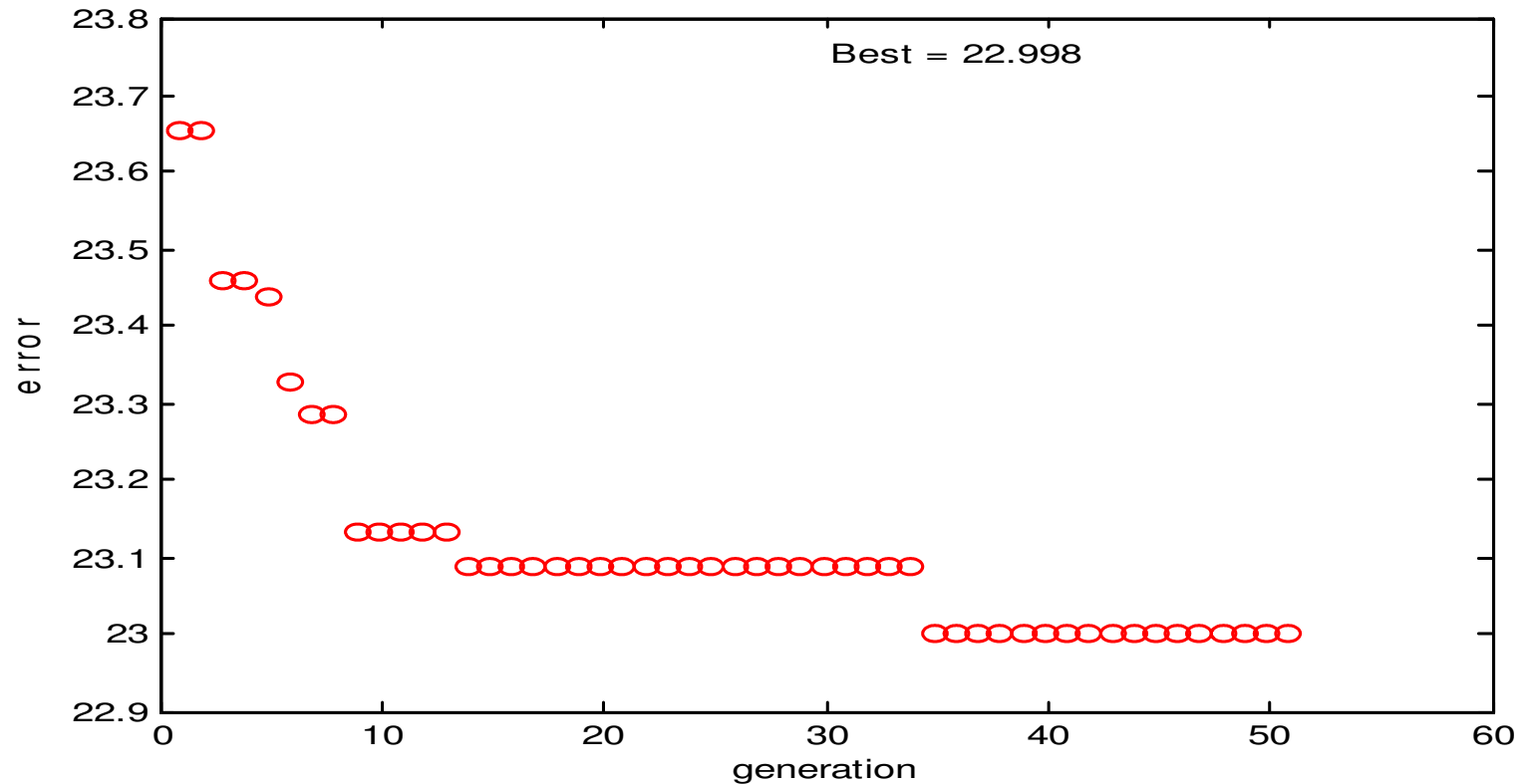
Taux d'erreur de l'hybridation GA-SVM-IS pour les **plosives**

Résultats sur « **Corpus TIMIT (a)** à 13 MFCC »



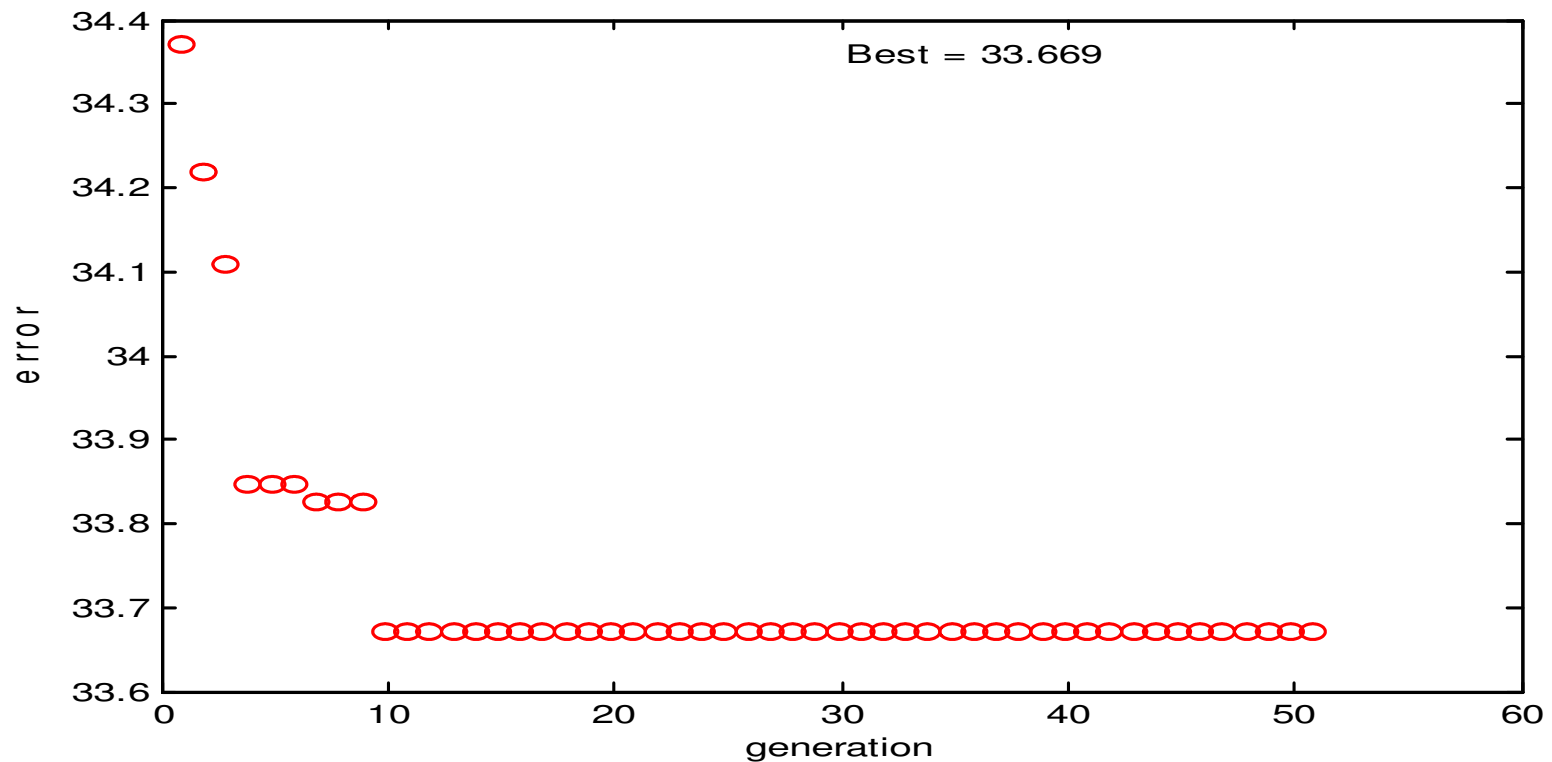
Taux d'erreur de l'hybridation **GA-SVM-FS** pour les
voyelles

Résultats sur « **Corpus TIMIT (a)** à 13 MFCC »



Taux d'erreur de l'hybridation GA-SVM-FS pour les **fricatives**

Résultats sur « **Corpus TIMIT (a)** à 13 MFCC »



Taux d'erreur de l'hybridation **GA-SVM-FS** pour les **plosives**

Résultats sur « **Corpus TIMIT (a)** » à 13 MFCC »

Paramètres Systèmes	Taux de reconnai- ssance(%))	Temps Train (Sec.)	Temp s Test (Sec.)	Complexité du modèle		Taille de l'ensemble d'apprentissa ge
				# VS	Taille de Matrice GRAM	
SVM	71.74	9.80 +15.49 +11.59	003 +0.01 +0.05	30 20 25	7571 x 7571 11887 x 11887 12056 x 12056	(7571, 416) (11887, 299) (12056, 286)
GA-SVM-IS	72.90	5.39 +5.09 +4.02	0.01 +0.03 +0.02	33 26 26	3759 x 3759 5931 x 5931 6063 x 6063	(3759 , 416) (5931 , 299) (6063 , 286)
GA-SVM-FS	72.18	11.38 + 9.22 + 7.42	0.05 +0.03 +0.05	27 18 21	7571 x 7571 11887 x 11887 12056 x 12056	(7571, 286) (11887, 195) (12056, 221)

Résultats sur « **Corpus TIMIT (b)** à 39 MFCC »

Paramètres Système	Taux de reconnai- -ssance (%)	Temps Train (Sec.)	Temps Test (Sec.)	Complexité du modèle		Taille de l'ensemble d'apprentissage
				# VS	Taille de Matrice GRAM	
SVM	73.02	1156.3 6	0.53	281	37694 x 37694	Ens. App. d'origine (37694, 1677)
GA-SVM-IS	76.22	654.41	0.25	195	19789 x 19789	Sous-ens3. (19789 , 1677)
GA-SVM-FS	74.62	704.57	0.27	176	37694 x 37694	Sous-ens4. (37694, 1053)

Résultats sur « **Corpus TIMIT (b)** » à 39 MFCC »

Paramètres Système	Taux de reconnaissance (%)	Temps Train (Sec.)	Temps Test (Sec.)	Complexité du modèle		Taille de l'ensemble d'apprentissage
				# VS	Taille de Matrice GRAM	
SVM	80.00	712.30 +239.86 +328.03	008 +0.05 +0.16	358 223 174	7739 x 7739 12519 x 12519 17436 x 17436	(7739,1677) (12519,1287) (17436,1209)
GA-SVM-SI	80.96	542.20 +193.56 +286.04	0.05 +0.04 +0.09	320 193 166	4267 x 4267 6951 x 6951 9086 x 9086	(4267 , 1677) (6951 ,1287) (9086 , 1209)
GA-SVM-FS	80.22	602.50 + 213.46 +315.84	0.05 +0.04 +0.09	310 179 158	7739x 7739 12519x 12519 17436x 17436	(7739, 1521) (12519, 1092) (17436, 1131)

Comparaison globale

Paramètre Approche	Approches utilisés	Taux de reconnaissance (%)		Obs.
		Global	Par cl-phoné.	
[Benyousef, 05]	BP AG-BP	- 59.46	(38, 45, 55) (62, 83, 69)	
[Senoussaoui, 05]	STM Td-STM Ts-STM SVM Td-STM & SVM	60.57 64.14 64.19 55.71 64.21	- - - -	
[Tlemsani, 05]	RPGC NN-GA NN-ES	- - -	46.78 52.39 58.81	
[Neggaz, 07]	NN-GA NN-ES NN-DE NN-MPGA NN-CSA	52.98 59.24 62.11 60.11 76.53	52.39 58.81 60.67 64.20 74.93	
Nos approches	SVM GA-SVM-IS GA-SVM-FS SVM (+Temps) GA-SVM-IS (+Temps) GA-SVM-FS (+Temps)	58.30 59.24 58.42 73.02 76.22 74.62	71.74 72.90 72.18 80.00 80.96 80.22	+ le gain important réalisé en termes temps d'apprentissage & simplicité du modèle de classification.

Conclusions & Perspectives

Contributions

- ✓ **Optimisation du SVM** réalisée de 2 manières différentes : IS & FS avec AG.
- ✓ Premier but : **accélération de la phase d'apprentissage** --> les résultats dans cet axe sont encourageants.
- ✓ **Diminuer la complexité** du modèle utilisé pour la classification.

Conclusions

- ✓ Introduction de l'**aspect temporel** au niveau des vecteurs MFCC est très rentable;
- ✓ Temps d'apprentissage d'un classificateur SVM varie **lentement** avec la dimension des vecteurs d'entrée, mais il est **plus sensible** au nombre de ces vecteurs
IS est **plus rentable** que la **FS** (Dans notre cas ?);
- ✓ Performances d'un système de RF dépendent de la phase d'**apprentissage** et de la **qualité** & la **quantité** de données disponibles par classe.

Perspectives

- ✓ Réalisation IS & FS, en utilisant d'autres méthaheuristiques tels que les colonies de fourmis, etc.
- ✓ Utilisation IS & FS avec d'autres classifieurs tq : les RB et/ou les RN.

Articles acceptés (en présentation Orale)

- **Optimisation des Machines à Vecteurs de Support Multiclasse par les Algorithmes Génétiques : Application à la classification phonétique;**

2^{èmes} Journées Doctorales en TIC (JDTIC'10), 15-17 Juillet 2010, Fès-Maroc.

- **Feature Selection for Multiclass Support Vector Machines;**

International Educational Technology Conference (IETC 2011); May 25-27, 2011, Istanbul, Turkey.

MERCI

