ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE DE LA PRODUCTION AUTOMATISÉE
M.Ing.

PAR
PHILIPPE HENNIGES

PSO POUR L'APPRENTISSAGE SUPERVISÉ DES RÉSEAUX NEURONAUX DE TYPE FUZZY ARTMAP

MONTRÉAL, LE 25 JUILLET 2006

CE MÉMOIRE A ÉTÉ ÉVALUÉ
PAR UN JURY COMPOSÉ DE :

Éric Granger, directeur de mémoire
Génie de la production automatisée, École de technologie supérieure

Robert Sabourin, codirecteur de mémoire
Génie de la production automatisée, École de technologie supérieure

Richard Lepage, professeur, président du jury
Génie de la production automatisée, École de technologie supérieure

Luiz Eduardo S. Oliveira, examinateur externe
Programa de Pós-Graduação em Informática Aplicada (PPGIA), Pontifícia Universidade Católica do Paraná (PUCPR)

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC
LE 27 Juin 2006
À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

PSO POUR L'APPRENTISSAGE SUPERVISÉ DES RÉSEAUX NEURONAUX DE TYPE FUZZY ARTMAP

Philippe Henniges

SOMMAIRE

Dans ce mémoire, nous avons étudié les divers comportements d'un type de réseau de neurones en particulier, soit le réseau fuzzy ARTMAP (FAM), dans le but de développer une stratégie d'apprentissage spécialisée pour ce type de réseau. Pour ce faire, nous avons observé les effets de plusieurs caractéristiques sur ce type de réseau, soit: la taille de la base de données d'entraînement, les stratégies d'apprentissage standard, la technique de normalisation, la structure du chevauchement, la polarité du MatchTracking ainsi que l'influence des paramètres internes du réseau fuzzy ARTMAP. Ces effets sont mesurés au niveau de la qualité et des ressources utilisées par le réseau FAM à travers des bases de données synthétiques et réelles.

Nous avons remarqué que le réseau FAM présente une dégradation de performances due à un effet de sur-apprentissage créé par le nombre de patrons d'entraînement et le nombre d'époques d'apprentissage, et ce, avec les bases de données possédant un degré de chevauchement. Pour éviter ce problème, nous avons développé une stratégie d'apprentissage spécialisée pour les réseaux FAM. Celle-ci permet d'améliorer les performances en généralisation en utilisant l'optimisation par essaims particulaires ou PSO (anglais pour "Particle Swarm Optimization") pour optimiser la valeur des quatre paramètres internes FAM (α, β, ε et $\overline{\rho}$).

Cette stratégie spécialisée obtient lors de toutes nos simulations, tant avec les bases de données synthétiques que réelles, de meilleures performances en généralisation que lors de l'utilisation des stratégies d'apprentissage standard utilisant les paramètres standard des réseaux FAM (MT+, MT-). De plus, elle permet d'éliminer la majorité de l'erreur de sur-apprentissage due à la taille de la base d'entraînement et au nombre d'époques d'apprentissage. Ainsi, cette stratégie spécialisée pour FAM a démontré que la valeur des paramètres internes du réseau FAM a un impact considérable sur les performances du réseau. De plus, pour toutes les bases testées, les valeurs optimisées des paramètres sont généralement toutes éloignées de leurs valeurs standard (MT- et MT+), lesquelles sont majoritairement utilisées lors de l'emploi du réseau FAM.

Cependant, cette stratégie d'apprentissage spécialisée n'est pas consistante avec la philosophie « on-line » de la famille ART, car la valeur des paramètres n'est pas optimisée séquentiellement. Malgré tout, elle permet d'indiquer les zones de performances optimales pouvant être atteintes par le réseau fuzzy ARTMAP. À notre connaissance, c'est la première fois qu'une stratégie d'apprentissage pour FAM utilise l'optimisation des valeurs des quatre paramètres internes de ce réseau.

PSO SUPERVISED LEARNING OF FUZZY ARTMAP NEURAL NETWORKS

Philippe Henniges

ABSTRACT

The impact on fuzzy ARTMAP neural network performance of decisions taken for batch supervised learning is assessed through computer simulations performed with different pattern recognition problems, like the hand writing numerical characters problem. To do so, we will study the impact of many characteristics on this neural network, such as: the training set size, the training strategies, the normalisation technique, the overlapping structure, the MatchTracking polarity and the impact of the fuzzy ARTMAP parameters. By allowing this network to learn real and synthetic data under various conditions, the extent of performance degradation is compared in terms of generalisation error and resources requirements.

Degradation of fuzzy ARTMAP performance due to overtraining is shown to depend on factors such as the training set size, and the number of training epochs, and occur for pattern recognition problems in which class distributions overlap. As an alternative to the commonly-employed hold-out training strategy, a strategy based on Particle Swarm Optimization (PSO), which determines both network parameters and weights such that generalisation error is minimized, has been introduced.

Through a comprehensive set of simulations, it has been shown that when fuzzy ARTMAP uses the PSO training strategy it produces a significantly lower generalisation error than when it uses typical training strategies. Furthermore, the PSO strategy eliminates degradation of generalisation error due to overtraining resulting from the training set size, number of training epochs, and data set structure. Overall results obtained with the PSO strategy highlight the importance of optimizing parameters (along with weights) for each problem, using a consistent objective function. In fact, the parameters found using this strategy vary significantly according to, *e.g.*, training set size and data set structure, and always differ considerably from the popular choice of parameters that allows to minimize resources.

The PSO strategy is inherently a batch learning mechanism, and as such is not consistent with the ARTMAP philosophy in that parameters cannot be adapted on-the-fly, through on-line, supervised or unsupervised, incremental leaning. Nonetheless, it reveals the extent to which parameter values can improve generalisation error of fuzzy ARTMAP, and mitigate the performance degradation caused by overtraining. To the best of our knowledge, it is the first time that a training strategy is developed for optimizing the four parameters of fuzzy ARTMAP neural network.

REMERCIEMENTS

La réalisation de ce mémoire a été possible grâce à la participation financière de M. Éric Granger et de M. Robert Sabourin. Je n'aurai pu accomplir cette tâche sans cette aide, et je tiens à les en remercier. Je tiens également à les remercier de leur constant soutien tout au long de la réalisation de ce mémoire, ainsi que de la confiance qu'ils ont placée en moi.

Merci à M. Tony Wong pour le support au niveau du besoin d'ordinateurs de calcul, cela à été grandement apprécié. Je tiens également à remercier M. Dominique Rivard de son aide et des longues heures où nous avons discuté et expérimenté divers problèmes reliés aux réseaux fuzzy ARTMAP.

Un immense merci à ma mère et à mon père pour m'avoir accompagné et encouragé tout au long de ce parcours.

Je veux également remercier M. Luiz Oliveira de nous avoir soumis quelques-unes de ses idées, notamment sur l'utilisation de l'algorithme d'optimisation par essaims particulaires (PSO). Finalement, merci à tous les membres du LIVIA pour leur soutien moral et les nombreuses discussions accompagnées de café.

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS ET DES SIGLES

FAM	réseau de neurones fuzzy ARTMAP
I	patrons d'entrées
$\mathbf{W}_i$	poids synaptique de la $i^{ème}$ catégorie
T_j	taux de ressemblance entre **I** et $\mathbf{W}_i$
$\cap$	ET booléen
$\wedge$	ET flou
α	paramètre de choix du réseau FAM
β	paramètre de la vitesse d'apprentissage du réseau FAM
ε	paramètre du MatchTracking du réseau FAM
$\bar{\rho}$	paramètre de vigilance de base du réseau FAM
$\mathbf{W}^{ab}$	poids synaptique activé sur la couche F2
$\mathbf{F}^{ab}$	lien entre la couche F2 et le "mapfield"
A	patron d'entrées complémenté
a_i	$i^{ème}$ caractéristique du patron **A**
a_i^c	$i^{ème}$ caractéristique complémentée du patron **A**
N	nombre de catégories
m	nombre de caractéristiques non complémentées d'un patron
M	nombre de caractéristiques complémentées d'un patron **A**
E_{gen}	erreur en généralisation
1EP	stratégie d'apprentissage typique avec une époque
$CONV_P$	stratégie d'apprentissage typique avec la convergence des patrons
$CONV_W$	stratégie d'apprentissage typique avec la convergence de poids synaptiques
HV	stratégie d'apprentissage typique avec la validation hold-out
PSO	optimisation par essaims particulaires *(Particle Swarm Optimization)*
gbest	particule ayant obtenu la meilleure *qualité*

$\overline{\mathbf{x}}_i$	position de la $i^{ième}$ particule
$G(\overline{\mathbf{x}}_i)$	*qualité* obtenue à la position de la $i^{ième}$ particule
$\overline{p}_g$	position de la $g^{ième}$ particule ayant obtenu la meilleure *qualité*
$G(\overline{p}_g)$	*qualité* obtenue à la meilleure position de la $g^{ième}$ particule
$\overline{\mathbf{v}}_i$	vitesse de la $i^{ième}$ particule
$\mathbf{V}_{max}$	vitesse maximum de déplacement des particules
$\mathbf{X}_{min}$	position minimum des particules
$\mathbf{X}_{max}$	position maximum des particules
$w(t)$	poids inertiel à la $t^{ième}$ itération PSO
PSO(1EP)	stratégie d'apprentissage spécialisée avec une époque
PSO($CONV_P$)	stratégie d'apprentissage spécialisée avec la convergence des patrons
PSO($CONV_W$)	stratégie d'apprentissage spécialisée avec la convergence des poids synaptiques
PSO(HV)	stratégie d'apprentissage spécialisée avec la validation hold-out
CQB	classificateur quadratique Bayesien
*k*NN	classificateur utilisant la règle des k plus proches voisins(k NeirestNetwork)
CRéduite	normalisation Centré Réduite
MinMax	normalisation MinMax
MPI	Message Passing Interface
$\overline{E}_{sapp}$	erreur de sur-apprentissage
MT-	MatchTracking négatif (ε = -0.001)
MT+	MatchTracking positif (ε = +0.001)
$a'_{i,k}$	valeur normalisée de la $i^{ème}$ caractéristique du $k^{ième}$ patron
$a_{i,k}$	valeur non normalisée de la $i^{ème}$ caractéristique du $k^{ième}$ patron
min_i	valeur minimale de la de la $i^{ème}$ caractéristique

max_i	valeur maximale de la de la i^{ème} caractéristique
μ_i	valeur moyenne de la i^{ème} caractéristique de l'ensemble des patrons utilisés lors de la phase d'apprentissage
σ_i	variance de la i^{ème} caractéristique de l'ensemble des patrons utilisés lors de la phase d'apprentissage
STD_{dev}	déviation standard
C	taux de compression du réseau fuzzy ARTMAP
$\lvert BD_{app} \rvert$	taille de la base d'apprentissage
$Nb_{catégories}$	nombre de catégories engendrées par un réseau fuzzy ARTMAP
E_{sapp}	erreur de sur-apprentissage

INTRODUCTION

La reconnaissance de formes a pour but d'identifier la classe d'un objet. Selon les applications, ces formes peuvent être des images, des signaux, des sons, etc. En mesurant des caractéristiques sur une forme (avec un capteur) nous transformons cette forme en un vecteur d'entré (patron ou observation).

La reconnaissance de formes a une longue histoire. Avant les années 60, elle était une branche de recherche de la mathématique statistique, mais comme beaucoup d'autres domaines scientifiques, cette discipline a pris un tournant avec l'arrivée des ordinateurs dans les centres universitaires. La reconnaissance de formes utilise souvent l'intelligence artificielle pour effectuer la classification des patrons. Aujourd'hui, l'intelligence artificielle utilisée pour la reconnaissance de formes est présente dans un grand nombre de domaines de recherche, allant de la robotique au médical, en passant par un grand nombre d'autres branches. Ces domaines d'applications ont aidé à développer et à améliorer le domaine de l'intelligence artificielle, lequel regroupe trois familles de techniques, soit les réseaux de neurones, les systèmes experts et les systèmes flous.

Mais qu'est-ce que l'intelligence artificielle? En fait, nous sommes encore très loin d'une entité pensante dotée d'une intelligence équivalente à l'humain, le tout résidant dans un ordinateur. Les logiciels d'intelligence artificielle d'aujourd'hui sont en fait des systèmes ayant la capacité de créer leurs propres règles de décision permettant la classification d'un objet. Bien entendu, le tout en plus compliqué. Pour pouvoir parler d'intelligence, il faut que le système apprenne de lui-même et non qu'un humain lui dicte toutes les actions à entreprendre et comment les effectuer. Il faut également faire attention de ne pas confondre un système expert avec un réseau de neurones. Un système expert est une application logicielle composée de deux entités: soit une base de connaissances qui a été enrichie par un cogniticien (spécialiste humain dans le domaine de compétence concerné) et un moteur d'inférence constitué d'un algorithme logiciel qui analyse les faits

d'un problème à l'aide de la base de connaissances. Un réseau de neurones est un modèle de calcul dont la conception est schématiquement inspirée du fonctionnement de vrais neurones (humains ou non) et dont l'apprentissage se fait sans interaction humaine. Pour qu'un réseau de neurones artificiels apprenne de lui-même, il doit posséder un minimum de connaissances innées. Ces connaissances innées sont des règles mathématiques concernant l'apprentissage et la classification. Il s'agit en fait de notre interprétation de ces mêmes commandes écrites dans les cellules grises de l'humain, de leurs mécanismes et de leurs fonctionnements.

Dans ce mémoire, nous allons ainsi étudier les divers comportements d'un type de réseau de neurones en particulier, soit le réseau fuzzy ARTMAP (FAM), dans le but de développer une stratégie d'apprentissage spécialisée pour ce type de réseau. Pour ce faire, nous allons observer les forces et les faiblesses du réseau FAM lors du traitement des divers problèmes. Nous allons étudier les effets de plusieurs caractéristiques sur ce type de réseau, telle la taille de la base de données d'entraînement, les stratégies d'apprentissage ainsi que l'influence des paramètres internes du réseau FAM. Ces effets seront mesurés au niveau de la qualité et des ressources utilisées par le réseau FAM à travers des bases de données synthétiques et réelles. Les effets de ces caractéristiques nous permettront d'optimiser les performances de classification de ce réseau lors de la reconnaissance de chiffres manuscrits.

Les effets de la taille de la base d'entraînement permettront d'établir si les réseaux FAM peuvent subir une dégradation des performances due à un trop grand nombre de patrons d'apprentissage. Ce phénomène survient dans les réseaux de neurones artificiels lorsqu'un réseau apprend trop précisément une base de données et perd sa capacité de généralisation. Ce phénomène est également présent chez l'humain. Rappelez-vous l'adage: « il est dur de bien réapprendre quelque chose de mal appris ». L'avantage avec les réseaux de neurones est que si l'on détecte ce phénomène, il suffit de recommencer le tout avec un réseau vierge alors que l'humain, lui, doit d'abord désapprendre ce qu'il a

mal appris. Les stratégies d'apprentissage servent à choisir le critère d'arrêt de l'apprentissage à l'intérieur du réseau de neurones. Les effets de ces stratégies permettent d'établir si les réseaux FAM peuvent également subir des pertes de performances dues à un apprentissage trop long.

Le but de ce mémoire est de développer une stratégie d'apprentissage spécialisée pour le réseau FAM permettant d'améliorer les performances en généralisation. Pour ce faire des bases de données synthétiques ainsi qu'une base de données réelles représentant les chiffres manuscrits de 0 à 9 sont utilisées. Les divers effets étudiés sur le réseau FAM ont permis de développer cette stratégie d'apprentissage spécialisée, laquelle optimise la valeur des quatre paramètres internes FAM lors de l'entraînement du réseau. Cette stratégie a permis d'améliorer de façon significative les performances obtenues par le réseau FAM sur la classification des chiffres manuscrits ainsi que sur les bases de données synthétiques.

Ce mémoire est divisé en cinq chapitres. Le chapitre 1 présente les notions théoriques utiles pour la compréhension de notre stratégie d'apprentissage spécialisée pour le réseau FAM. Le chapitre 2 traite de la méthodologie expérimentale utilisée pour accomplir les simulations effectuées et ainsi permettre à un tiers de réitérer nos expérimentations afin d'avoir la possibilité de faire la preuve du concept. Les chapitres 3 et 4 présentent respectivement les résultats obtenus par les bases de données synthétiques lors de l'utilisation des stratégies d'apprentissage standard ainsi que ceux obtenus lors de l'utilisation des stratégies d'apprentissage spécialisées pour les réseaux FAM. Le chapitre 5 expose tous les résultats obtenus avec la base de données réelles NIST SD19. Finalement, une conclusion récapitule les divers résultats obtenus par toutes les expérimentations et recommande des pistes intéressantes pour de futurs travaux.

CHAPITRE 1

OPTIMISATION PAR ESSAIMS PARTICULAIRES AVEC FUZZY ARTMAP

Lors de la réalisation de ce mémoire, plusieurs méthodes ont été utilisées. Ce chapitre présente les divers aspects théoriques pour bien comprendre les expériences effectuées tout au long de ce mémoire.

Ce chapitre est divisé en trois sections. La première section présente la théorie du classificateur neuronique FAM. Il s'agit du classificateur à la base de ce mémoire et une bonne compréhension de son fonctionnement interne est nécessaire. La deuxième section présente les techniques d'apprentissage standard et l'effet de sur-apprentissage. Puis, la dernière section présente la stratégie d'apprentissage spécialisée utilisant l'optimisation par essaims particulaires (PSO) que nous avons développée. Cette stratégie utilise l'algorithme PSO afin d'éliminer la dégradation des performances présente dans ces réseaux.

1.1 Réseau de neurones fuzzy ARTMAP

Le fuzzy ARTMAP est un réseau de neurones artificiels appartenant à la famille ARTMAP qui permet l'apprentissage supervisé et non supervisé des observations dont les caractéristiques sont définies dans l'espace des nombres réels. Le principe de base des réseaux de neurones de type FAM a été introduit par Carpenter, Grossberg, Markuzon, Reynolds et Rosen en 1992 [2]. Ce type de réseau exploite le réseau fuzzy ART [6] qui utilise un algorithme non supervisé pour la génération des groupes (clustering). En fait, ce type de réseau est obtenu en remplaçant les modules ART1 [3,4] du système ARTMAP par un module fuzzy ART [5,6]. Le module fuzzy ART est obtenu en remplaçant l'opérateur d'intersection de type booléen du module ART1 par un opérateur ET de type flou. Le Tableau I présente les différences au niveau algorithmique

entre le module ART1 exploité par le modèle ARTMAP et le module fuzzy ART exploité par le modèle FAM.

Tableau I

Algorithme ART1 vs fuzzy ART [2]

ART1	fuzzy ART
FONCTION DE CHOIX	
$T_j = \frac{\lvert \mathbf{I} \cap \mathbf{w}_j \rvert}{\alpha + \lvert \mathbf{w}_j \rvert}$	$T_j = \frac{\lvert \mathbf{I} \wedge \mathbf{w}_j \rvert}{\alpha + \lvert \mathbf{w}_j \rvert}$
FONCTION DE VIGILANCE	
$\frac{\lvert \mathbf{I} \cap \mathbf{w} \rvert}{\lvert \mathbf{I} \rvert} \geq \rho$	$\frac{\lvert \mathbf{I} \wedge \mathbf{w} \rvert}{\lvert \mathbf{I} \rvert} \geq \rho$
FONCTION D'APPRENTISSAGE	
$\mathbf{w}_j^{(new)} = \mathbf{I} \cap \mathbf{w}_j^{(old)}$	$\mathbf{w}_j^{(new)} = \mathbf{I} \wedge \mathbf{w}_j^{(old)}$
$\cap$ = ET booléen (intersection)	$\wedge$ = ET flou (minimum)

Le FAM est simple à utiliser car un petit nombre de paramètres est nécessaire pour sa mise en oeuvre. En effet, son comportement est basé principalement sur le paramètre de choix (α), le paramètre de vigilance de base ($\bar{\rho}$), de MatchTracking (ε) et le paramètre d'apprentissage (β). Sa complexité algorithmique est faible et peut être mise en œuvre très efficacement en électronique numérique. Il existe d'autres algorithmes qui s'inspirent de la logique floue (eg, [7,8,]), mais le FAM se distingue de ceux-ci par le fait qu'il modifie ses poids synaptiques après chaque observation (apprentissage

séquentiel) au lieu d'effectuer un apprentissage "batch" après avoir inspecté l'ensemble des observations disponibles. Ceci est un avantage notable pour les applications d'apprentissage dites en-ligne. La figure 1 représente l'architecture du modèle FAM.

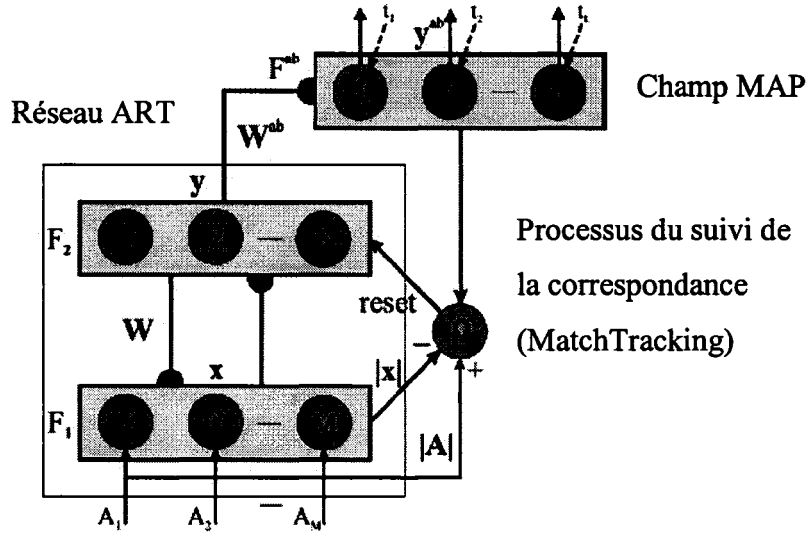


Figure 1 Architecture du Fuzzy ARTMAP [32]

Le réseau fuzzy ARTMAP contient trois couches distinctes, soit la couche F_1, F_2 et F^{ab} (le champ MAP). La valeur des caractéristiques du patron **A** est propagée à la couche F_1. La couche F_1 produit le vecteur d'activation $\mathbf{x} = \mathbf{A} \wedge \mathbf{W}$ et est reliée à la couche F_2 par les poids synaptiques **W**. La couche F_2 représente les catégories créées dans le réseau par lequel le patron est classifié. Elle est reliée au champ MAP par les poids $\mathbf{W}^{ab}$ et produit le vecteur d'activation **y**. Le champ MAP produit le vecteur d'activation $\mathbf{y}^{ab}$. En fait, il active une classe *K* selon la catégorie de la couche F_2 sélectionnée. Lors de la phase d'apprentissage, le champ MAP active, si besoin, le processus du suivi de la correspondance (MatchTracking), qui utilise les valeurs du patron **A** et de la couche F_1 (**x**) afin de modifier la recherche de la catégorie appropriée pour le patron présenté.

A - Phase d'apprentissage :

1 - Initialisation :

Lors de l'initialisation du réseau, aucun nœud de la couche F_2 n'est assigné. Les entrées (**a**, **t**) sont présentées au réseau, où **a** représente le vecteur de données des observations et **t** la classe qui lui est associée. Le réseau FAM exige que toutes les caractéristiques a_i des observations présentées au système soient codées en complément et qu'elles soient toutes comprises entre 0 et 1 inclusivement. Le complément d'un patron de m dimensions est composé de $M = 2 \cdot m$ dimensions et est défini par :

$$\mathbf{A} = (a; a^c) = \left(a_1, a_2, ..., a_m; a_1^c, a_2^c, ..., a_m^c\right) \tag{1.1}$$

$$a_i^c = (1 - a_i) \tag{1.2}$$

$$a_i \in [0,1] \tag{1.3}$$

2 - Activation des catégories :

L'observation **A** active la couche F_1 et elle est propagée vers F_2 à travers les connexions synaptiques de poids **W**. L'activation des nœuds j de la couche F_2 est déterminée selon la loi de Weber :

$$T_j(\mathbf{A}) = \frac{\left|\mathbf{A} \wedge \mathbf{w}_j\right|}{\alpha + \left|\mathbf{w}_j\right|}, \quad j = 1,2,..N \tag{1.4}$$

$$\left|\mathbf{w}_j\right| = \sum_{i=1}^{M} w_i \tag{1.5}$$

Le paramètre de choix α possède une valeur supérieure ou égale à 0. Ce paramètre influence le nombre d'itérations de recherche des catégories avant d'assigner une nouvelle catégorie déterminant ainsi la profondeur de recherche de la couche F_2. Ainsi, une grande valeur de ce paramètre favorise la création de nouvelles catégories alors qu'une petite valeur favorise la réutilisation des catégories existantes. La couche F_2 sélectionne un seul gagnant J = argmax $\{T_j ; j=1,2,..,N\}$. Ainsi, seulement le nœud J, ayant la plus grande valeur d'activation, restera actif. Si plus d'un nœud a la même valeur d'activation et qu'elle est maximale, le nœud avec le plus petit indice sera

sélectionné. Le test de vigilance s'exécute alors avec le nœud *J*. Il compare le degré de similitude entre $\mathbf{w}_J$ et **A** avec le paramètre de vigilance ρ :

$$\frac{|\mathbf{A} \wedge \mathbf{w}_j|}{M} \geq \rho \tag{1.6}$$

Si le test est concluant, l'activation des classes survient, sinon, le nœud *J* est désactivé et le système cherche un autre nœud respectant le test de vigilance. Si aucun nœud ne satisfait l'équation (1.6), un nœud de la couche F_2 non assigné est attribué à cette entrée. Le paramètre de vigilance détermine la taille maximale des catégories, soit $|w_j| \leq 2 \cdot (1-\rho)$ et sa valeur est comprise entre 0 et 1. De plus, ce paramètre influence le nombre de catégories créées. Une grande valeur de ce paramètre favorise la création de nouvelles catégories. À l'opposé, une petite valeur de ce paramètre tend à réutiliser le plus grand nombre de catégories possibles et ainsi diminue le nombre de catégories créées.

3 - Activation des classes :

Lorsque la couche F_2 sélectionne le nœud *J* gagnant, elle active le champ MAP via les poids synaptiques $\mathbf{W}^{ab}$ et assigne une classe à l'observation présentée. Si la réponse de ce nœud n'est pas la même que celle associée au patron d'entrée (**t**) , le processus du suivi de la correspondance (MatchTracking) s'enclenche. Ce processus augmente la valeur du paramètre de vigilance ρ (équation (1.7)) dans le but d'effectuer une nouvelle recherche à travers la couche F_2. Celle-ci continuera jusqu'à ce qu'un nœud classifie correctement l'observation présentée, ou jusqu'à qu'un nœud non assigné soit associé à ce patron.

$$\rho = \frac{|\mathbf{A} \wedge \mathbf{w}_J|}{|\mathbf{A}|} + \varepsilon \tag{1.7}$$

La valeur du paramètre de MatchTracking ε peut être négative ou positive. Une grande valeur positive favorise la création de nouvelles catégories lorsque la première tentative de classification d'un patron échoue. Une grande valeur négative facilite la réutilisation des catégories déjà existantes lorsque la classification a initialement échoué et ainsi tend à minimiser le nombre de catégories créées lors de l'apprentissage.

4 - Apprentissage :

L'apprentissage d'une observation **a** met à jour les poids synaptiques $\mathbf{w}_J$, et crée un nouveau lien associatif vers F^{ab} si J correspond à un nœud de la couche F_2 nouvellement associé. La mise à jour des poids synaptiques de la couche F_2 est effectuée selon l'équation :

$$w_J^{'} = \beta(A \wedge w_J) + (1-\beta) \cdot w_J \tag{1.8}$$

où : β est la vitesse d'apprentissage

Cet algorithme peut être ajusté pour un apprentissage lent, $0 < \beta << 1$, ou pour un apprentissage rapide, $\beta = 1$. Lors de l'utilisation du mode d'apprentissage rapide, le réseau FAM formera des hyper-rectangles de largeur minimale pour entourer l'observation **a**. Une petite valeur du paramètre β diminue la vitesse de changement de la taille des catégories alors qu'une grande valeur permet un changement de taille de plus en plus rapide.

B - Phase de test :

Lors de la phase de test, l'observation présentée au réseau est associée à un nœud de la couche F_2 par la fonction de choix. La classe du champ d'association du nœud gagnant est associée avec l'observation testée, classant ainsi cette observation. Aucun test de vigilance ou de MatchTracking n'est effectué.

1.2 Stratégies d'apprentissage standard et impact du sur-apprentissage

Un réseau de neurones nécessite une phase d'apprentissage par laquelle le réseau construit les liens lui permettant d'effectuer la classification du type de données présentées. Lors de cette phase, le réseau doit posséder un critère d'arrêt d'apprentissage. Si le réseau apprend trop longtemps, un phénomène de sur-apprentissage risque de survenir, entraînant une dégradation des performances en généralisation. Le sur-apprentissage est un phénomène provenant de la sur-spécialisation d'un système pour un problème donné. Ainsi, dans un réseau de neurones, à partir d'un certain nombre d'époques[1] d'entraînement, le système se sur-spécialise par rapport à la base de données d'apprentissage. Ce faisant, il perd sa capacité de généralisation par rapport aux données de test provenant de la même source mais qui n'ont pas encore été traitées par le système. La figure 2 représente ce phénomène.

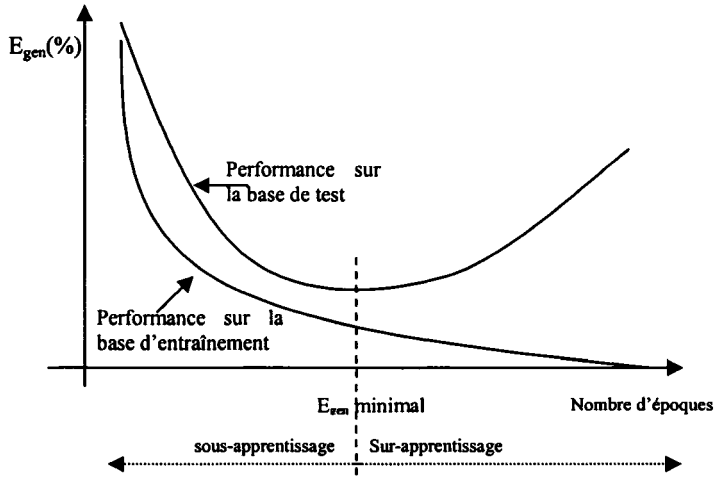


Figure 2 Schématisation de l'erreur en généralisation en fonction du nombre d'époques lors de la phase d'apprentissage d'une classification

[1] Une époque d'entraînement est atteinte lorsque tous les patrons contenus dans la base d'apprentissage ont été présentés au réseau.

Cette figure présente l'erreur en généralisation en fonction du nombre d'époques d'apprentissage. On constate que l'erreur en généralisation sur la base d'apprentissage ne fait que décroître. Ainsi, le réseau de neurones devient meilleur d'époque en époque pour classifier la base d'entraînement. Par contre, à partir d'un certain nombre d'époques, on remarque que l'erreur en généralisation sur la base de test augmente. Cet effet est dû au fait que le réseau s'est sur-spécialisé pour la base d'entraînement et qu'il perd sa capacité de généralisation. Ainsi, il faut arrêter l'entraînement du système à l'époque la plus proche possible de la frontière entre le sous-apprentissage et le sur-apprentissage. Avec les réseaux FAM, le sur-apprentissage se manifeste par une dégradation des performances en généralisation et en ressources. La stratégie d'apprentissage par validation hold-out permet de réduire les effets engendrés par le sur-apprentissage.

Cependant, le phénomène de sur-apprentissage est légèrement controversé dans la communauté du FAM. Boaz Lerner et Boaz Vigdor [18] obtiennent de meilleures performances lors de l'apprentissage par convergence des patrons d'entraînement comparativement à l'utilisation d'une méthode de validation appliquée sur le nombre d'époques d'apprentissage. En d'autres mots, il ressort de leur étude que le nombre d'époques d'entraînement ne crée pas de sur-apprentissage pour le réseau FAM. À l'opposé, deux équipes indépendantes ont démontré que le FAM subissait bel et bien une erreur de sur-apprentissage due au nombre d'époques d'entraînement [22,23,26].

Dans certaines situations, le FAM a tendance à créer plus de catégories que nécessaire [19]. Bref, il perd sa capacité de généralisation à cause de la création d'un trop grand nombre de poids synaptiques à l'intérieur du réseau. Par contre, ce phénomène survient même lors de l'application de la stratégie d'apprentissage validation hold-out, laquelle est immunisée contre la dégradation des performances due au nombre d'époques d'entraînement. Quelques solutions pour régler ce problème ont été proposées dans la communauté scientifique [19,20,21].

Il existe plusieurs stratégies pour indiquer à un réseau l'arrêt de l'apprentissage. Les quatre stratégies les plus courantes sont :

Une époque *(1EP)* : Entraînement du réseau sur une seule époque. Le seul critère d'arrêt de cette méthode est la durée de l'apprentissage, soit une époque. Avant l'apprentissage, l'ordre de présentation des données est généré aléatoirement.

Convergence des poids synaptiques *($CONV_W$)* : Entraînement du réseau avec comme condition d'arrêt la convergence des poids synaptiques. L'ordre de présentation des observations est réparti au hasard entre chaque époque. La convergence des poids synaptiques implique que la somme des différences au carré entre les poids obtenus par deux époques successives soit inférieure à 0,01. Ceci indique que les valeurs des poids obtenues entre deux époques n'ont pratiquement pas changé.

Convergence des patrons d'apprentissage *($CONV_P$)* : Entraînement du réseau avec comme condition d'arrêt la convergence des patrons d'apprentissage. L'ordre de présentation des observations est réparti au hasard entre chaque époque d'entraînement. La convergence des patrons d'apprentissage implique que la phase d'entraînement se termine lorsque toutes les observations de la base d'apprentissage sont parfaitement classées.

Validation hold-out *(HV)* : Pour résoudre le problème de sur-apprentissage, des méthodes de validation croisée ont été proposées [9,10,17]. La technique la plus courante est la méthode de validation hold-out. Cette méthode a été développée à partir des techniques de validation croisée, mais elle n'utilise qu'un seul ensemble de données pour la validation et n'utilise pas de « croisement » lors de la validation. La différence entre la validation croisée et la validation de type hold-out est importante car la validation croisée est supérieure pour des petits ensembles de données [30]. La validation hold-out implique la séparation de la base de données d'apprentissage en deux

parties distinctes et fixes : une base d'apprentissage et une base de validation. La base de validation servira de test après chaque époque d'apprentissage et permettra de sceller l'apprentissage du réseau lors de l'obtention du minimum d'erreur en généralisation (E_{gen} minimal), voir figure 2. Une fois l'apprentissage terminé (convergence des patrons de la base d'apprentissage) le réseau créé à l'époque ayant obtenu le meilleur taux de reconnaissance sur la base de validation est conservé. Sa performance est, par la suite, évaluée sur la base de test. La base de validation permet donc de faire un choix sur le nombre d'époques d'apprentissage et ainsi garantir que la base de test n'est traitée qu'une seule fois par le réseau, soit lors du test final. L'ordre de présentation des observations est réparti au hasard entre chaque époque.

1.3 Stratégie d'apprentissage avec optimisation par essaims particulaires

Cette section présente la stratégie d'apprentissage spécialisée pour les réseaux FAM que nous avons développée. Cette stratégie utilise PSO pour sélectionner la valeur des quatre paramètres internes FAM, soit le paramètre de choix α, de vigilance de base $\bar{\rho}$, de MatchTracking ε et de vitesse d'apprentissage β.

D'autres travaux ont été effectués pour étudier l'impact de certains paramètres. Le paramètre de vigilance a été testé avec diverses valeurs [18] ainsi qu'avec une valeur variable pendant l'entraînement [39,40]. Bien que ces auteurs ont démontré un avantage à utiliser une valeur de vigilance variable, l'effet secondaire de cette approche est une prolifération des catégories diminuant ainsi le taux de compression. Dubarwski [41,42] a utilisé une approche stochastique pour orienter l'apprentissage des réseaux neuronaux. Ses travaux portent sur les paramètres α, β et ρ des réseaux fuzzy ARTMAP. Les résultats obtenus sont raisonnablement comparables à ceux obtenus par un humain sur un problème à deux dimensions.

1.3.1 Optimisation par essaims particulaires

Plusieurs scientifiques ont créé des modèles interprétant le mouvement des vols d'oiseaux et des bancs de poissons. Plus particulièrement, Reynolds [11] et Heppner et al. [12] ont présenté des simulations sur un vol d'oiseaux. Reynolds était intrigué par l'aspect esthétique du déplacement des oiseaux en groupe et Hepnner, un zoologue, était intéressé à comprendre les règles permettant à un grand nombre d'oiseaux de voler en groupe : soit de voler sans se heurter, de changer soudainement de direction, de s'écarter et de se rapprocher de nouveau. Cette étude a grandement inspiré le développement de l'algorithme PSO.

En effet, lors de simulations des modèles mathématiques décrivant les vols d'oiseaux, Wilson [13] a suggéré que ces types de modèles pourraient très bien s'appliquer à la recherche de points caractéristiques dans un espace de recherche. Sa réflexion se base sur le fait que, lors de l'installation d'une mangeoire à oiseaux dans une cour, même si aucun oiseau ne l'a jamais visitée, après quelques heures de patience un grand nombre d'oiseaux viendront y manger. Lors des simulations de Wilson, la volée d'oiseaux cherchait une mangeoire dans un espace donné et finissait par découvrir son emplacement. En utilisant les algorithmes de modélisation de Heppner [12] et de Reynolds [11], et en modifiant le modèle mathématique de Wilson [13], Kennedy et Eberhart [14] ont transformé le tout en un vol d'oiseaux cherchant la « mangeoire » la plus grosse dans un lot de mangeoires contenues dans une région prédéterminée. L'algorithme d'optimisation PSO a ainsi vu le jour.

1.3.1.1 Description algorithmique

L'algorithme PSO est généralement introduit en relatant son développement conceptuel. Tel que mentionné, cet algorithme a vu le jour sous la forme d'une simulation simplifiée d'un milieu social, tel que le déplacement des oiseaux à l'intérieur d'une volée. Pour cet

algorithme une redéfinition des termes est nécessaire; une *population* représente le vol d'oiseaux et un *agent ou particule* représente chaque oiseau de la volée.

Lors de l'initialisation de PSO, chaque particule est positionnée aléatoirement dans l'espace de recherche. Pour définir ce positionnement il faut connaître la plage des valeurs de chaque dimension (paramètres) à optimiser. Puis, la vitesse initiale de chaque particule est aléatoirement choisie. Pour choisir cette vitesse, l'algorithme requiert une vitesse maximale absolue pour chaque dimension. La vitesse peut être négative mais le signe n'indique que le sens dans lequel la particule se déplace.

Il est fortement conseillé d'initialiser la première particule à l'aide d'une valeur connue [15]. Lors de l'optimisation des paramètres dans l'espace de recherche, ceux-ci possèdent généralement des valeurs par défaut. Ces valeurs deviennent le point de départ de la recherche. L'initialisation de la première particule avec de telles valeurs permet d'obtenir, dès la première itération, une valeur de pertinence *(qualité)* de base. Cette valeur peut servir de référence pour fins de comparaison et d'estimation de la performance en optimisation obtenue à l'aide de PSO.

Chaque particule connaît son emplacement (x_i), sa vitesse de déplacement (v_i), la position où elle a obtenu sa meilleure qualité (P_i) ainsi que la position de la meilleure qualité obtenue par l'ensemble des particules (P_g) et ce, à chaque itération (t) de l'algorithme PSO. En tenant compte de ces valeurs, la position et la vitesse de déplacement de chaque particule est mise à jour. La figure 3 présente la mise d'une particule avec l'algorithme PSO.

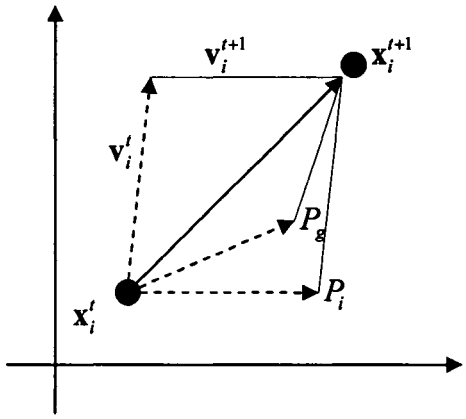


Figure 3 Mise à jour d'une particule avec l'algorithme PSO

Le but de l'algorithme PSO est d'optimiser une fonction continue dans un espace donné. Dans la majorité des cas, l'algorithme d'optimisation recherche le maximum ou le minimum global de l'espace de recherche. Voici la description des étapes de l'algorithme PSO :

Étape 0 :

Si le critère d'arrêt est vérifié, alors l'algorithme se termine. S'il ne l'est pas, une nouvelle itération commence en retournant à l'Étape 1 avec la première particule $(i=1)$.

Le critère d'arrêt correspond généralement à un nombre d'itérations prédéfinies, mais on peut également spécifier un critère d'arrêt en fonction de la meilleure valeur de qualité $G(\overline{p}_g)$ obtenue pour l'ensemble des particules.

Pour toutes les particules de la population, exécuter les Étapes 1 à 5.

Étape 1 :

Calcul de la qualité $G(\overline{\mathbf{x}}_i)$ de la particule i en fonction de son vecteur de position $(\overline{\mathbf{x}}_i)$.

Étape 2 :

Établir si la qualité $G(\overline{\mathbf{x}}_i)$ obtenue par la particule *i* est supérieure à la meilleure qualité que cette particule a obtenue antérieurement. Si $G(\overline{\mathbf{x}}_i) > G(\overline{p}_i)$, la présente position de la particule $\overline{\mathbf{x}}_i$ est sauvegardée comme étant la meilleure position $\overline{p}_i$ obtenue à ce jour pour la particule *i*.

Étape 3 :

Établir si la qualité $G(\overline{p}_i)$ obtenue par la particule *i* est plus grande que la meilleure qualité $G(\overline{p}_g)$ obtenue pour l'ensemble de la population. Si tel est le cas, l'indice de la particule ayant obtenu la meilleure qualité *g* prend la valeur *i*.

Étape 4 :

Mettre à jour la vitesse de déplacement $\overline{\mathbf{v}}_i(t)$ de la particule *i*. Cette mise à jour tient compte de la vitesse précédente de la particule $\overline{\mathbf{v}}_i(t-1)$, de sa position présente $(\overline{\mathbf{x}}_i)$, de la position de la meilleure qualité $\overline{p}_i$ obtenue par cette particule ainsi que de la position de la meilleure qualité globale $\overline{p}_g$ obtenue par la population. De plus, deux paramètres, φ_1 et φ_2, sont utilisés pour ajuster l'importance des termes $(p_{id} - x_{id}(t-1))$ et $(p_{gd} - x_{id}(t-1))$ de l'équation de mise à jour de la vitesse (1.11). Une fois cette vitesse mise à jour, il faut vérifier si la nouvelle vitesse $\overline{\mathbf{v}}_i(t)$ de la particule *i* est contenue dans les limites autorisées $\overline{\mathbf{v}}_i \in (-\mathbf{V}_{max}, +\mathbf{V}_{max})$. Si tel n'est pas le cas, la nouvelle vitesse est réduite à la borne la plus proche.

Étape 5 :

Mettre à jour la position $\overline{\mathbf{x}}_i(t)$ de la particule *i*. Cette mise à jour tient compte de la position précédente de la particule $\overline{\mathbf{x}}_i(t-1)$ ainsi que de la nouvelle vitesse $\overline{\mathbf{v}}_i(t)$ calculée à l'étape 4. Une fois la position de la particule *i* mise à jour, il faut vérifier si la

nouvelle position $\overline{\mathbf{x}}_i(t)$ est contenue dans l'espace de recherche spécifié par $\overline{\mathbf{x}}_i \in (\mathbf{X}_{\min}, \mathbf{X}_{\max})$. Si tel n'est pas le cas, la nouvelle position est ramenée à la borne la plus proche.

L'algorithme 1 présente un sommaire de l'algorithme PSO tel que présenté par J. Kennedy et R. Ebernhart [14,15].

Algorithme 1 : Optimisation par essaims particulaires [14,15]

Initialisation des paramètres $\overline{\mathbf{x}}_i, \overline{\mathbf{v}}_i, \mathbf{V}_{\max}, \mathbf{X}_{\min}, \mathbf{X}_{\max}$

for t = 1 au temps maximum

  for i = 1 au nombre de particules

    if $G(\overline{x}_i) > G(\overline{p}_i)$ //G() évaluer la qualité

      $\overline{\mathbf{p}}_i = \overline{\mathbf{x}}_i$ // p_{id} meilleure position

    EndIf

    g = i // arbitraire

    for j = index des voisins de la particule i

      If $G(\overline{\mathbf{p}}_j) > G(\overline{\mathbf{p}}_g)$ then g = j //index de la meilleure particule globale

    next j

    $\overline{\mathbf{v}}_i(t) = \overline{\mathbf{v}}_i(t-1) + \varphi_1(\overline{\mathbf{p}}_i - \overline{\mathbf{x}}_i(t-1)) + \varphi_2(\overline{\mathbf{p}}_g - \overline{\mathbf{x}}_i(t-1)), \quad \overline{\mathbf{v}}_i \in (-\mathbf{V}_{\max}, +\mathbf{V}_{\max})$

    $\overline{\mathbf{x}}_i(t) = \overline{\mathbf{x}}_i(t-1) + \overline{\mathbf{v}}_i(t), \quad x_i \in (\mathbf{X}_{\min}, \mathbf{X}_{\max})$

  next i

next t, jusqu'à obtention du critère d'arrêt

1.3.1.2 Valeur des paramètres φ_1 et φ_2

L'équation la plus importante dans l'algorithme PSO est la mise à jour de la vitesse de déplacement des particules $\overline{\mathbf{v}}_i(t)$:

$$\overline{\mathbf{v}}_i(t) = \overline{\mathbf{v}}_i(t-1) + \varphi_1(\overline{\mathbf{p}}_i - \overline{\mathbf{x}}_i(t-1)) + \varphi_2(\overline{\mathbf{p}}_g - \overline{\mathbf{x}}_i(t-1)) \quad (1.9)$$

Les valeurs attribuées à φ_1 et φ_2 ont fait l'objet d'une recherche. Après un très grand nombre de simulations et bien des tentatives, Kennedy et Eberhart [14] ont proposé un modèle simple pour ces deux valeurs. Ces deux variables seront d'une valeur aléatoire comprise entre 0 et 2 pour chaque mise à jour de chaque vitesse, et cela pour chaque particule (agent). Bien que plusieurs autres méthodes d'attribution des paramètres φ_1 et φ_2 ont été proposées, ce modèle reste le plus performant [14]. Ainsi, après modifications, la nouvelle formule de mise à jour de la vitesse des particules est :

$$\overline{\mathbf{v}}_i(t) = \overline{\mathbf{v}}_i(t-1) + \varphi_1(\overline{\mathbf{p}}_i - \overline{\mathbf{x}}_i(t-1)) + \varphi_2(\overline{\mathbf{p}}_g - \overline{\mathbf{x}}_i(t-1)) \quad (1.10)$$

Où : $\varphi_1 = 2 \cdot rand()$ et $\varphi_2 = 2 \cdot rand()$

1.3.1.3 Poids inertiel

Shi et Eberhart [27] ont modifié la formule de mise à jour de la vitesse en introduisant la notion de poids inertiel. Le poids inertiel (*w(t)*) est un paramètre utilisé pour équilibrer la force de la vitesse précédente de la particule. La formule de mise à jour de la vitesse d'une particule est maintenant :

$$\overline{\mathbf{v}}_i(t) = \overline{\mathbf{v}}_i(t-1) \cdot w(t) + 2 \cdot rand() \cdot (\overline{\mathbf{p}}_i - \overline{\mathbf{x}}_i(t-1)) + 2 \cdot rand() \cdot (\overline{\mathbf{p}}_g - \overline{\mathbf{x}}_i(t-1)) \quad (1.11)$$

De nombreux tests ont également été effectués pour trouver la valeur optimale de *w(t)* [16]. L'une des meilleures techniques trouvée consiste en une fonction linéaire diminuant la valeur de *w(t)* de 0.9 à 0.4 sur le nombre maximal d'itérations à effectuer. Ainsi, la valeur de *w(t)* diminue légèrement après chaque itération PSO. Au début d'une optimisation, les particules feront de grands déplacements. Ceci permettra d'explorer une grande partie de l'espace. Puis, à mesure que le nombre d'itérations augmente, la

grandeur des déplacements des particules diminuera, permettant ainsi de raffiner la recherche.

1.3.1.4 Nombre de particules

Le choix du nombre de particules utilisées lors de l'optimisation est un autre sujet traité par Kennedy et Eberhart [14]. Bien que le nombre de particules optimum soit variable selon les types de problèmes à optimiser, il est généralement conseillé d'utiliser de 10 à 30 particules lors de l'emploi de l'algorithme PSO. Pour améliorer le résultat de l'optimisation, il est également recommandé de faire quelques cycles PSO lors de l'utilisation d'un petit nombre de particules, car cet algorithme est sensible aux valeurs initiales des particules. On peut diminuer cette sensibilité en utilisant un grand nombre de particules (50 et plus) et ainsi n'exécuter qu'un seul cycle d'optimisation PSO.

1.3.2 Stratégie d'apprentissage spécialisée pour le fuzzy ARTMAP

En utilisant PSO nous avons développé une stratégie d'apprentissage spécialisée pour les réseaux FAM afin d'éliminer la dégradation des performances présente dans ces réseaux.

Nous savons que les quatre paramètres internes des réseaux FAM peuvent influencer les performances de ces réseaux et que la valeur par défaut de ces paramètres est presque toujours utilisée. En employant PSO, nous allons chercher de meilleures valeurs pour les quatre paramètres FAM dans le but d'obtenir de meilleures performances en généralisation. Ainsi, pour chaque problème de classification testé, notre stratégie d'apprentissage spécialisée, utilisant l'algorithme PSO, optimisera les performances en généralisation de FAM à l'aide d'une base de validation dans un espace de recherche de quatre dimensions (paramètre de choix α, de vigilance de base $\bar{\rho}$, de MatchTracking ε et de vitesse d'apprentissage β).

Les quatre stratégies d'apprentissage standard sont implémentées lors du calcul de la qualité des particules. Lors de l'utilisation de la stratégie d'apprentissage standard 1EP pour le calcul de la qualité, on dénotera cette stratégie d'apprentissage spécialisée : PSO(1EP). Les stratégies spécialisées utilisant les autres stratégies d'apprentissage standard sont dénotées: PSO($CONV_P$), PSO($CONV_W$) et PSO(HV). Il existe donc deux critères d'arrêt dans chaque stratégie d'apprentissage spécialisée. L'une des deux conditions d'arrêt provient de la stratégie d'apprentissage standard lors du calcul de la qualité de chaque particule. La seconde condition d'arrêt est propre à l'algorithme d'optimisation PSO et est basée sur le nombre d'itérations de recherche. Si après 10 itérations PSO aucune particule n'a obtenu une meilleure performance en généralisation (calcul de la qualité) que la meilleure obtenue globalement, la recherche est arrêtée. De plus, l'optimisation PSO est limitée à 100 itérations. Une fois la centième itération complétée, l'optimisation PSO s'arrêtera.

Lors de l'entraînement, les valeurs des quatre paramètres internes FAM seront limitées dans un espace de recherche. La méthodologie utilisée pour les stratégies spécialisées pour les réseaux FAM est décrite en détail à la section 2.2.1.

CHAPITRE 2

MÉTHODOLOGIE EXPÉRIMENTALE

Ce chapitre a pour but de présenter la méthodologie expérimentale utilisée lors des expériences effectuées pour mesurer l'impact des divers facteurs des réseaux fuzzy ARTMAP. Le protocole expérimental utilisé traite des aspects relatifs aux simulations effectuées dans le cadre de ce projet de recherche.

Ce chapitre contient six sections. La première section présente les bases de données utilisées, soit les bases de données synthétiques et réelles. La deuxième section traite de l'emploi des diverses stratégies d'apprentissage utilisées lors des simulations. La section suivante présente les algorithmes de référence utilisés pour comparer les performances obtenues par les réseaux FAM. La quatrième section traite des techniques de normalisation utilisées lors du prétraitement des données pour les réseaux FAM. La cinquième section décrit les mesures de performance effectuées lors des simulations. Finalement, la dernière section présente le banc de test et les logiciels utilisés pour la réalisation des simulations.

2.1 Bases de données

Cette section décrit les bases de données synthétiques et réelles utilisées pour l'ensemble des simulations.

2.1.1 Bases de données synthétiques

Toutes les bases synthétiques utilisées sont formées de deux classes à 2 dimensions, équiprobables, dont la surface de répartition des classes est identique. Chaque base de données est composée de 300k patrons, distribués également entre deux classes. Ces bases de données sont divisées en trois bases de 100k patrons, soit la base

d'apprentissage, de validation et de test, chacune distribuée également entre les deux classes. La Figure 4 représente cette division.

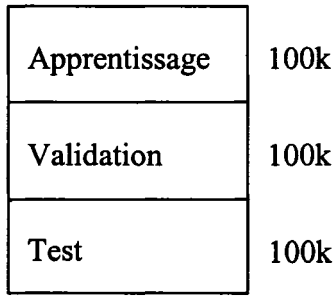


Figure 4 Séparation des bases de données synthétiques

Dans le but de présenter des résultats moyens face aux performances en généralisation du FAM, ces trois parties sont ensuite divisées en dix parties égales (BD_{APPi} BD_{VALIDi} BD_{TESTi} , *où* $i=[1,2,3,...,10]$) permettant dix réplications pour chaque problème. Chacune des 10 réplications est composée de trois bases de données de 10k patrons soit une base d'apprentissage, une base de validation et une base de test, chacune répartie également entre les deux classes.

Chaque réplication comporte 30 tests (S_i), chacun faisant augmenter graduellement la taille de la base d'apprentissage selon une progression logarithmique. Ainsi, le tout premier test (S_1) utilise les 10 premiers patrons (5 de chaque classe) de la première division de la base d'apprentissage (BD_{APP1}). Puis, le second test (S_2) utilise les 12 premiers patrons (6 de chaque classe) toujours de la première division de la base d'apprentissage (BD_{APP1}). Le tout jusqu'au $30^{ième}$ test qui utilise tous les patrons de la première division de la base d'apprentissage. Pour chacun de ces 30 tests, la base de test DB_{TEST1} est utilisée. Si une base de validation est nécessaire, DB_{VALID1} est utilisée. Finalement, le tout sera répété pour chacune des dix réplications.

Le Tableau II présente la taille des bases d'apprentissage utilisées lors des 30 tests effectués avec les bases de données synthétiques.

Tableau II

Taille des bases d'apprentissage pour les simulations avec les données synthétiques

TEST No	Taille de la base d'apprentissage	TEST No	Taille de la base d'apprentissage	TEST No	Taille de la base d'apprentissage
#1	10	#11	108	#21	1172
#2	12	#12	136	#22	1486
#3	16	#13	174	#23	1886
#4	20	#14	220	#24	2395
#5	24	#15	280	#25	3038
#6	32	#16	356	#26	3856
#7	40	#17	452	#27	4892
#8	52	#18	572	#28	6210
#9	66	#19	726	#29	7880
#10	84	#20	922	#30	10000

Quatre types différents de bases de données synthétiques sont utilisés. Ces quatre types de bases de données se définissent comme suit :

BD_{μ} - Base de données respectant une distribution normale, avec une frontière de décision linéaire, dont le degré de chevauchement est dû au rapprochement des moyennes des classes. La variance de chaque classe reste fixe.

BD_{σ} - Base de données respectant une distribution normale, avec une frontière de décision linéaire, dont le degré de chevauchement est dû à l'augmentation des variances des classes. La moyenne de chaque classe reste fixe.

BD_{CiS} - Base de données provenant du problème *circle in square* [1]. Ce problème possède une erreur théorique nulle dont la frontière de décision est complexe.

BD_{P2} - Base de données provenant du problème P_2 [28]. Ce problème possède une erreur théorique nulle dont les frontières de décision sont complexes.

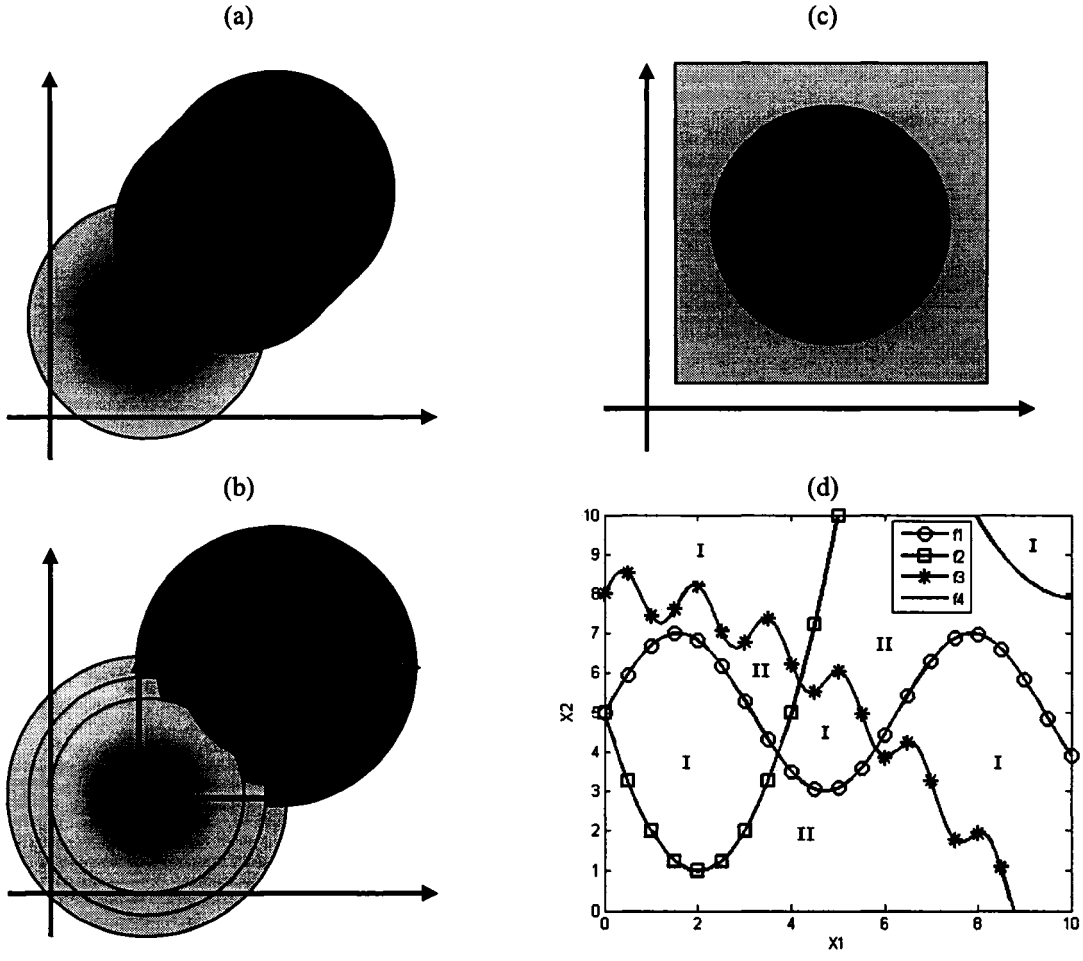


Figure 5 Représentation des bases de données synthétiques (a) DB_{μ}, (b) DB_{σ}, (c) DB_{CiS} et (d) DB_{P2}.

Les deux premiers types de bases de données représentent des problèmes dont le chevauchement est dû, respectivement, au rapprochement des moyennes des classes (DB_{μ}), et au rapprochement des variances des deux classes (DB_{σ}). Les deux derniers

types de bases de données (DB_{CIS} et BD_{P2}) représentent des problèmes de classification ne possédant aucun degré de chevauchement avec une ou des bornes de décision complexes. La Figure 5 présente ces quatre types de bases de données. Le degré de chevauchement des bases de données respectant une distribution normale (DB_μ et DB_σ) est graduellement augmenté de 1% à 25% d'erreur. Le Tableau III présente la valeur, au millième près, des paramètres utilisés pour la création des divers degrés de chevauchement lors du rapprochement des moyennes des classes (DB_μ). Le Tableau IV présente les paramètres utilisés pour la création des divers degrés de chevauchement lors de l'augmentation des variances des deux classes (DB_σ).

Tableau III

Paramètres des distributions normales pour les bases DB_μ

Probabilité d'erreur	μ_1	μ_2	σ_1^2	σ_2^2
$DB_\mu(\varepsilon = 1\%)$	(0, 0)	(3.290, 3.290)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 3\%)$	(0, 0)	(2.660, 2.660)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 5\%)$	(0, 0)	(2.326, 2.326)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 7\%)$	(0, 0)	(2.087, 2.087)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 9\%)$	(0, 0)	(1.896, 1.896)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 11\%)$	(0, 0)	(1.735, 1.735)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 13\%)$	(0, 0)	(1.593, 1.593)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 15\%)$	(0, 0)	(1.466, 1.466)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 17\%)$	(0, 0)	(1.349, 1.349)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 19\%)$	(0, 0)	(1.242, 1.242)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 21\%)$	(0, 0)	(1.141, 1.141)	(1, 1)	(1, 1)

Probabilité d'erreur	μ_1	μ_2	σ_1^2	σ_2^2
$DB_\mu(\varepsilon = 23\%)$	(0, 0)	(1.045, 1.045)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 25\%)$	(0, 0)	(0.954, 0.954)	(1, 1)	(1, 1)

Tableau IV

Paramètres des distributions normales pour les bases DB_σ

Probabilité d'erreur	μ_1	μ_2	σ_1^2	σ_2^2
$DB_\sigma(\varepsilon = 1\%)$	(0, 0)	(3.290, 3.290)	(1, 1)	(1, 1)
$DB_\sigma(\varepsilon = 3\%)$	(0, 0)	(3.290, 3.290)	(1.530, 1.530)	(1.530, 1.530)
$DB_\sigma(\varepsilon = 5\%)$	(0, 0)	(3.290, 3.290)	(2.000, 2.000)	(2.000, 2.000)
$DB_\sigma(\varepsilon = 7\%)$	(0, 0)	(3.290, 3.290)	(2.485, 2.485)	(2.485, 2.485)
$DB_\sigma(\varepsilon = 9\%)$	(0, 0)	(3.290, 3.290)	(3.011, 3.011)	(3.011, 3.011)
$DB_\sigma(\varepsilon = 11\%)$	(0, 0)	(3.290, 3.290)	(3.597, 3.597)	(3.597, 3.597)
$DB_\sigma(\varepsilon = 13\%)$	(0, 0)	(3.290, 3.290)	(4.266, 4.266)	(4.266, 4.266)
$DB_\sigma(\varepsilon = 15\%)$	(0, 0)	(3.290, 3.290)	(5.038, 5.038)	(5.038, 5.038)
$DB_\sigma(\varepsilon = 17\%)$	(0, 0)	(3.290, 3.290)	(5.944, 5.944)	(5.944, 5.944)
$DB_\sigma(\varepsilon = 19\%)$	(0, 0)	(3.290, 3.290)	(7.022, 7.022)	(7.022, 7.022)
$DB_\sigma(\varepsilon = 21\%)$	(0, 0)	(3.290, 3.290)	(8.322, 8.322)	(8.322, 8.322)
$DB_\sigma(\varepsilon = 23\%)$	(0, 0)	(3.290, 3.290)	(9.914, 9.914)	(9.914, 9.914)
$DB_\sigma(\varepsilon = 25\%)$	(0, 0)	(3.290, 3.290)	(11.90, 11.90)	(11.90, 11.90)

La Figure 6 présente les frontières de décision entre les classes de la base DB_{P2} originale [28]. Les fonctions mathématiques décrivant ces frontières sont présentées par les équations (2.1) à (2.4).

$$f_1(x) = 2 * \sin(x) + 5 \tag{2.1}$$

$$f_2(x) = (x-2)^2 + 1 \tag{2.2}$$

$$f_3(x) = -0.1 \cdot x^2 + 0.6\sin(4x) + 8 \tag{2.3}$$

$$f_4(x) = \frac{(x-10)^2}{2} + 7 \tag{2.4}$$

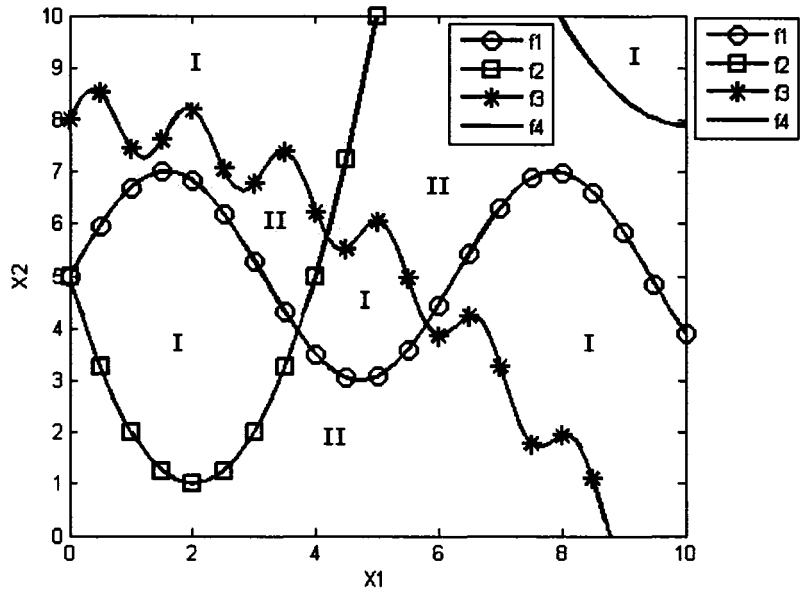


Figure 6 Frontière des classes de la base DB_{P2} [28]

Bien que la distribution de la base de données DB_{P2} soit uniforme et qu'il y ait le même nombre d'observations appartenant aux deux classes, la surface occupée par les observations des deux classes n'est pas identique. Le Tableau V présente le pourcentage de la surface occupée par chacune des classes de la base DB_{P2}.

Tableau V

Surface occupée par chaque classe de la base DB_{P2} originale

	Classe 1 *(I)*	Classe 2 *(II)*
Surface	52.2098%	47.7902 %

Pour obtenir la même probabilité *a priori* des deux classes, une légère modification de l'équation $f_4(x)$ (2.4) a permis d'obtenir une surface égale pour les observations des deux classes. La nouvelle fonction $f_4(x)$ est décrite par l'équation (2.5).

$$f_4(x) = \frac{(x-10)^2}{2} + 7.902 \tag{2.5}$$

2.1.2 Base de données réelles

La base de données réelles utilisée est la base NIST SD19. Elle est composée de 814255 images représentant des chiffres manuscrits (0 à 9). Cette base est divisée en huit sections *hsf*_{0,1,2,3,4,6,7,8}. La création de la base d'apprentissage et des bases de validation s'est effectuée en regroupant les bases de données *hsf*_{0,1,2,3}. Deux bases de test sont également utilisées : la base hsf_7 *(60 089 patrons)* étant la base de test standard de NIST SD19 et la base hsf_4 *(58 646 patrons)* étant une base bruitée, augmentant ainsi la difficulté de la classification.

La base de données NIST n'est pas une base de données équilibrée, ce qui veut dire que le nombre de patrons appartenant à une classe n'est pas égal à l'intérieur d'une série *hsf* de la base NIST SD19. Le Tableau VI présente le nombre de patrons contenus dans chacune des séries utilisées de la base NIST. Les séries *hsf*_{0,1,2,3} ont été

rééquilibrées car elles sont utilisées lors de la phase d'apprentissage alors que les séries *hsf*_{4,7}, utilisées pour la phase de test, ont été laissées telles quelles. Le nombre total d'images utilisées pour la base d'apprentissage et les trois bases de validation est de 195 000, soit 19 500 par classe. À partir de ces images, 132 caractéristiques sont extraites modélisant les aspects de chaque image. Ces caractéristiques ont été extraites par M. Oliveira lors de son Ph.D. à l'École de technologie supérieure de Montréal en collaboration avec le laboratoire d'imagerie, de vision et d'intelligence artificielle (LIVIA) [29]. Elles représentent des ratios de concavité (78 caractéristiques), de contour (48 caractéristiques) et de surface (6 caractéristiques) [29]. Ainsi, dans les simulations avec la base NIST SD19, 313 735 patrons, composés de 132 caractéristiques, représentant 10 classes, sont utilisés. Le Tableau VII présente la répartition de ces patrons à travers les diverses bases de données utilisées.

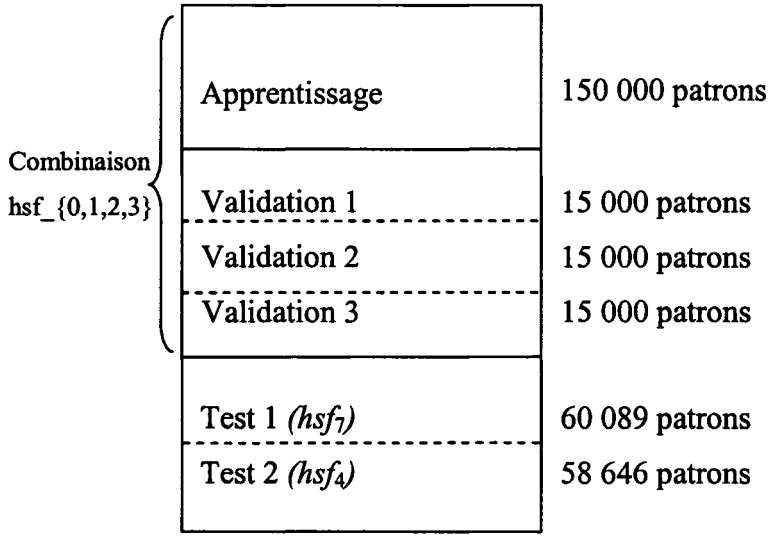
Tableau VI

Répartition des patrons de la base NIST SD19 dans les séries hsf

Class	*hsf*_{0,1,2,3}	hsf_4	hsf_7
0	22,971	5,560	5,893
1	24,772	6,655	6,567
2	22,131	5,888	5,967
3	23,172	5,819	6,036
4	21,549	5,722	5,873
5	19,545	5,539	5,684
6	22,128	5,858	5,900
7	23,208	6,097	6,254
8	22,029	5,695	5,889
9	21,619	5,813	6,026
Total	223,124	58,646	60,089

Tableau VII

Séparation des données de la base NIST SD19



Dans le but de présenter des résultats moyens face aux performances en généralisation du FAM, les simulations sont répétées 10 fois avec différents ordres de présentation définis aléatoirement. Chaque réplication comporte 15 tests provenant de la division de la base de données d'apprentissage en 15 tranches, où chaque tranche augmente graduellement la taille de la base d'apprentissage selon une règle logarithmique. Le premier test de la base NIST SD19 utilise une base de données d'apprentissage de 100 patrons (10 patrons par classe). Puis le deuxième test utilise une base de données d'apprentissage de 160 patrons (16 patrons par classe), et ainsi de suite jusqu'à ce que la base d'apprentissage soit de 150k patrons (15k par classe).

Le Tableau VIII présente les 15 grandeurs de la base d'apprentissage utilisées pour les simulations avec la base de données réelles.

Tableau VIII

Augmentation de la taille de la base d'apprentissage avec les données réelles

TEST No	Taille de la base d'apprentissage	TEST No	Taille de la base d'apprentissage	TEST No	Taille de la base d'apprentissage
#1	100	#6	1360	#11	18560
#2	160	#7	2290	#12	31290
#3	280	#8	3870	#13	52760
#4	470	#9	6520	#14	88960
#5	800	#10	11000	#15	150000

2.2 Stratégies d'apprentissage

Quatre stratégies d'apprentissage standard sont appliquées pour l'ensemble des bases de données. Ces quatre méthodes d'apprentissage sont :

a. Une époque (1EP);
b. Convergence des poids synaptiques ($CONV_W$);
c. Convergence des patrons d'apprentissage ($CONV_P$);
d. Validation hold-out (HV).

Le fonctionnement de ces méthodes est décrit à la section 1.2. De plus, nous avons développé une technique d'apprentissage spécifique pour les réseaux FAM. Cette technique utilise un algorithme d'optimisation afin d'améliorer les performances en généralisation du réseau FAM en sélectionnant de nouvelles valeurs pour les quatre paramètres internes du FAM. Cette technique utilise l'algorithme PSO pour la phase d'optimisation et une méthode d'apprentissage typique pour calculer la qualité. Ainsi, quatre techniques d'apprentissage spécialisées sont utilisées avec les quatre stratégies d'apprentissage standard et l'algorithme PSO. Ces quatre méthodes sont:

a. Optimisation PSO avec une époque "PSO(1EP)";
b. Optimisation PSO avec convergence des poids "PSO($CONV_W$)";
c. Optimisation PSO avec convergence des patrons d'apprentissage "PSO($CONV_P$)";
d. Optimisation PSO avec validation hold-out "PSO(HV)".

2.2.1 Stratégie d'apprentissage spécialisée avec optimisation par essaims particulaires

Lors des simulations avec les stratégies d'apprentissage spécialisées pour FAM utilisant PSO, l'algorithme PSO utilise 15 particules de recherche pour trouver la meilleure performance en généralisation sur l'ensemble des paramètres à optimiser. Les quatre paramètres internes des réseaux FAM sont optimisés soit: le choix (α), la vigilance ($\bar{\rho}$), le taux d'apprentissage (β) et le MatchTracking (ε). Ces paramètres sont optimisés pour chaque test, chaque base de données et chaque stratégie d'apprentissage. La plage d'optimisation utilisée pour chacun de ces paramètres est :

a. α : [0.00001, 1];
b. $\bar{\rho}$: [0, 1];
c. β : [0, 1];
d. ε : [-1, 1].

La vitesse maximale d'évolution d'une particule selon chaque paramètre est :

a. Vmax(α) : 0.1;
b. Vmax($\bar{\rho}$) : 0.1;
c. Vmax(β) : 0.1;
d. Vmax(ε) : 0.2.

L'évaluation de la performance de la première particule est effectuée à partir des valeurs par défaut utilisées avec les stratégies standard, soit: $\alpha = 0.01$, $\overline{\rho} = 0.0$, $\beta = 1.0$ et $\varepsilon = 0.001$. Toutes les autres particules sont initialisées aléatoirement sur la plage d'optimisation de chaque paramètre. Le nombre maximum d'itérations PSO est de 100 et quatre cycles d'optimisation sont effectués pour chaque expérience afin de minimiser l'impact de l'initialisation des particules. Il est à noter que l'algorithme PSO n'a jamais atteint la limite de 100 itérations lors de l'optimisation de tous les problèmes que nous avons testés.

Tel que décrit par l'algorithme 1 (voir section 1.3.1.1), au cours de l'optimisation PSO lorsqu'un réseau FAM obtient une meilleure valeur de qualité comparée à la meilleure qualité globale (*gbest)*, ce réseau devient le nouveau *gbest*. Lors des expériences d'optimisation avec les réseaux neuroniques FAM, nous avons remarqué que plusieurs réseaux différents obtiennent exactement les mêmes performances en généralisation. En utilisant cette connaissance à notre avantage, nous avons ajouté une nouvelle règle lors de l'établissement du meilleur réseau global :

> si un réseau obtient une valeur de qualité égale à celle de *gbest*, le réseau ayant la plus grande compression, soit le moins de catégories, est sélectionné comme étant l'optimum global (*gbest*).

Lors d'un cycle d'optimisation, lorsque que 10 itérations successives n'ont pas réussi à trouver un nouveau *gbest*, le cycle est arrêté. Le réseau FAM ayant obtenu la meilleure performance en généralisation sur la base de validation lors des 4 cycles est sélectionné comme étant le réseau optimal. Si plus d'un réseau a la même performance en généralisation, le réseau ayant le moins de catégories, soit le plus grand taux de compression, est sélectionné. Finalement, le meilleur réseau neuronique FAM trouvé lors des 4 cycles d'optimisation PSO est testé sur la base de test.

Les quatre stratégies d'apprentissage standard sont utilisées pour le calcul de la qualité à l'intérieur de l'algorithme PSO. Afin d'obtenir les meilleures performances possibles, l'ordre de présentation est également réparti au hasard entre chaque époque. Par contre, étant donné la grande demande en ressources pour l'optimisation des paramètres internes du réseau FAM, l'optimisation PSO est accomplie uniquement pour les bases de données $DB_{\mu}(1\%)$, $DB_{\mu}(9\%)$, $DB_{\mu}(25\%)$, $DB_{\sigma}(9\%)$, DB_{P2}, DB_{CIS} et uniquement avec la technique de normalisation MinMax.

Les stratégies d'apprentissage spécialisées pour FAM utilisant PSO sont également appliquées sur la base de données réelles NIST SD19. Étant donné la grande demande en temps de calcul de cet algorithme d'optimisation, une seule méthode d'apprentissage est utilisée sur cette base de données. Cette méthode est sélectionnée à partir des résultats obtenus par les tests effectués sur les données synthétiques.

2.3 Algorithmes de référence

Afin de bien étudier les résultats obtenus par les réseaux FAM, leurs performances sont comparées avec d'autres algorithmes de classification. Deux algorithmes différents sont utilisés comme référence, soit le classificateur quadratique Bayesien et la règle des *k* plus proches voisins. Malheureusement, ces deux classificateurs ne peuvent servir de référence dans certains types de problèmes. Le classificateur quadratique Bayesien est utilisé uniquement lors des simulations avec les bases DB_{μ} et DB_{σ} comme référence.

Il existe plusieurs avantages d'utiliser des données synthétiques, soit, entre autre, la connaissance parfaite de la distribution des classes. Grâce à cette connaissance il est possible de calculer l'erreur théorique ($\varepsilon_{théorique}$) pour chaque type de problème. Ainsi, les performances obtenues par les réseaux FAM et les performances des algorithmes de référence sont également comparées à l'erreur théorique de chaque type de base de données.

2.3.1 Classificateur quadratique Bayesien

Lors de l'utilisation de bases de données synthétiques respectant une distribution normale, le classificateur Bayesien est utilisé pour obtenir une performance de référence face aux résultats des simulations, et ce avec les bases DB_{μ} et DB_{σ}. Son utilisation comme algorithme de référence est basé sur le fait que ces bases de données *(DB_{μ} et DB_{σ})* respectent une distribution normale des données pour chaque classe.

Ce classificateur utilise une base de données d'apprentissage afin de calculer sa performance en généralisation. Cependant, l'erreur minimale théorique provenant de la connaissance des paramètres (moyenne et variance) des distributions normales n'est atteinte que si la base de données d'apprentissage tend vers l'infini. Quelques logiciels existent pour calculer l'erreur Bayesienne. Nous avons utilisé la plateforme PRTOOLS [31] de Robert P.W. Duin. L'annexe 1 présente le fonctionnement du classificateur Bayesien.

2.3.2 La règle du k plus proches voisins (kNN)

Lors des simulations avec *k*NN, le choix du nombre de voisins sur lequel s'effectue le vote pour déterminer la classification d'un patron de test est important. Plutôt que de fixer un nombre de voisins k pour toutes les simulations, le paramètre k est optimisé pour chaque simulation. Ainsi, chaque simulation vérifiera cinq différentes valeurs de k (1,3,5,7,9) sur la base validation, puis la valeur de k ayant obtenu la meilleure performance sera sélectionnée pour effectuer le test sur la base de test.

Le *k*NN est utilisé comme algorithme de référence pour toutes les bases de données synthétiques et réelles. L'annexe 2 présente le fonctionnement du classifieur *k*NN.

2.4 Normalisation des bases de données

Le réseau FAM requiert que toutes les caractéristiques des patrons présentés au réseau soient comprises dans l'intervalle [0, 1] inclusivement. Pour respecter cette spécificité du FAM, un prétraitement doit être effectué sur les bases de données. Ce prétraitement s'appelle la normalisation des données. Pour comprendre l'impact engendré suite à la méthode de normalisation utilisée, deux techniques de normalisation sont testées pour toutes les simulations effectuées, soit: la technique de normalisation MinMax et la technique de normalisation Centrée Réduite.

La normalisation MinMax est décrite par l'équation (2.6). Cette méthode de normalisation linéaire garantit que 100% des données normalisées sont comprises dans l'intervalle [0, 1]. Cependant, elle possède un désavantage. Si l'ensemble des données à normaliser comprend une donnée aberrante (donnée dont la fréquence d'occurrence est beaucoup moins élevée que toutes les autres) les données non aberrantes se retrouveront fortement compressées les unes sur les autres.

$$a'_{i,k} = \frac{a_{i,k} - \min_i}{\max_i - \min_i} \tag{2.6}$$

Où : $a'_{i,k}$ représente la valeur normalisée de la $i^{ème}$ caractéristique du $k^{ième}$ patron

$a_{i,k}$ représente la valeur non normalisée de la $i^{ème}$ caractéristique du $k^{ième}$ patron

$\min_i$ représente la valeur minimale de la de la $i^{ème}$ caractéristique

$\max_i$ représente la valeur maximale de la de la $i^{ème}$ caractéristique

La normalisation Centrée Réduite (CRéduite) est décrite par l'équation (2.7). Cette méthode de normalisation linéaire garantit que 68% des données ($[\mu-\sigma, \mu+\sigma]$) seront normalisées dans l'intervalle [-1, 1]. Pour obtenir un taux de 99% des données normalisées dans cet intervalle, le dénominateur de l'équation (2.7) doit être remplacé

par $3\sigma_i$. Une fois les données normalisées par l'équation (2.7), il faut exécuter une translation des données de l'intervalle [-1, 1] à [0, 1].

Lors de l'utilisation de la méthode de normalisation Centrée Réduite, les données non comprises dans l'intervalle [-1, 1] sont mises à l'extremum (1 ou -1) le plus proche.

$$a'_{i,k} = \frac{a_{i,k} - \mu_i}{\sigma_i} \tag{2.7}$$

Où : μ_i représente la valeur moyenne de la $i^{ème}$ caractéristique de l'ensemble des patrons utilisés lors de la phase d'apprentissage

σ_i représente la variance de la $i^{ème}$ caractéristique de l'ensemble des patrons utilisés lors de la phase d'apprentissage

2.4.1 Bases de données synthétiques et réelles

Lors de l'utilisation des bases de données synthétiques, les deux techniques de normalisation sont applicables car les données ne sont pas définies dans l'intervalle [0, 1]. Avec la base de données réelles, toutes les caractéristiques sont déjà comprises dans l'intervalle [0, 1]. Les 132 caractéristiques extraites des images représentent des ratios de concavité, de contour et de surface [29].

Cependant, il est possible d'effectuer une normalisation des données, même si celles-ci sont déjà comprises dans l'intervalle [0, 1] et ce, afin d'améliorer la dispersion des données sur cet intervalle. En regardant les histogrammes de chaque caractéristique de la base NIST SD19, nous constatons que ces données pourraient bénéficier d'une normalisation, car elles sont majoritairement situées près de 0. La figure 8 présente l'histogramme de la $15^{ième}$ caractéristique extraite à partir de la base NIST SD19, lequel est représentatif de la majorité des 132 caractéristiques.

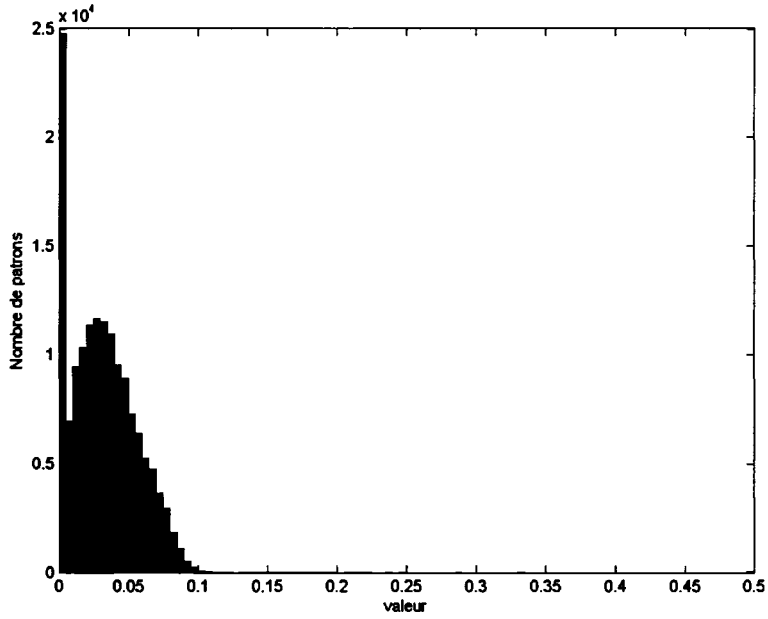


Figure 7 NIST SD19, caractéristique #15

Le chapitre 4 présente une discussion sur l'impact des deux techniques de normalisation appliquées sur les bases de données synthétiques. Le chapitre 5 contient une section portant sur les forces et les faiblesses des deux techniques de normalisation appliquées sur la base de données réelles NIST.

2.5 Mesures de performance

Pour chaque simulation, des mesures de performance sont effectuées. Ces mesures représentent la capacité de généralisation ainsi que les ressources utilisées par les réseaux FAM. La qualité du réseau FAM est obtenue par la mesure de l'erreur en généralisation et les ressources utilisées par le réseau FAM sont mesurées par le temps de convergence et le taux de compression.

Dans chaque cas, une mesure de dispersion est appliquée pour connaître la variabilité statistique des résultats. Pour mesurer la dispersion des résultats, l'équation de la déviation standard utilisée en mathématique statistique (2.8) est utilisée.

$$STD_{dev} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}} \tag{2.8}$$

Où : n est la taille de la population

x_i est la valeur du ième individu en cours d'évaluation

$\overline{x}$ est la moyenne de la population

L'erreur en généralisation

L'erreur en généralisation est le rapport entre le nombre d'observations d'une base de test dont la classification obtenue est incorrecte sur le nombre total d'observations contenues dans cette base. Toutes les données de la base de test ne servent qu'une seule fois, soit pour le test final.

Le temps de convergence

Le temps de convergence montre le nombre d'époques d'entraînement qui ont été nécessaires au réseau FAM lors d'une simulation avant l'obtention de la condition d'arrêt. Une époque d'apprentissage signifie que toutes les observations de la base d'apprentissage ont été soumises au réseau.

Le taux de compression du réseau

Le taux de compression obtenu par le réseau FAM est calculé selon la formule (2.9).

$$C = \frac{\left|BD_{app}\right|}{Nb_{catégories}} \tag{2.9}$$

Où : $\left|BD_{app}\right|$ est la taille de la base d'apprentissage

$Nb_{catégories}$ est le nombre de catégories engendrées par le réseau

2.6 Banc de test

Le banc de test utilisé pour ces expérimentations a été créé et testé dans le cadre du laboratoire de recherche LIVIA de l'École de technologie supérieure de Montréal. Pour obtenir la convivialité du banc de test, le logiciel MatLAB est utilisé. Par contre, la performance de MatLAB au niveau de la rapidité de traitement laisse quelque peu à désirer. Pour obtenir un niveau de performance optimal, les calculs effectués par le réseau FAM sont codés en C. Ainsi, le banc de test se divise en deux grandes sections, soit la section MatLAB codée par M. Philippe Henniges et la section du code C codée par M. Dominique Rivard. La figure 1.7 présente le schéma des échanges de haut niveau du banc de test.

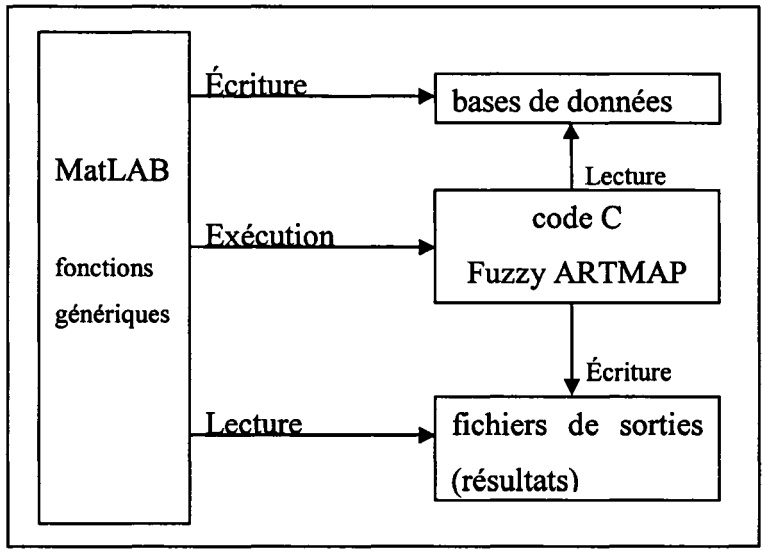


Figure 8 Schéma des échanges de haut niveau du banc de test

Bien que le noyau du code C du réseau FAM soit identique pour toutes les méthodes d'apprentissage, un exécutable a été compilé pour chaque méthode d'apprentissage. L'algorithme PSO a été codé sous MatLAB et il appelle la fonction C de son choix pour réaliser l'apprentissage d'un réseau FAM sur une base de données.

Lors des simulations PSO avec la base de données réelles NIST SD19, les simulations ont été exécutées sur une grappe d'ordinateurs (Beowulf cluster). L'algorithme PSO synchrone a été utilisé pour paralléliser le processus d'optimisation en utilisant 15 nœuds. Pour effectuer ces simulations, la création d'un nouveau code C utilisant les méthodes de traitement en parallèle MPI a été nécessaire.

De plus, la plateforme PRTOOLS [31] de Robert P.W. Duin a été utilisée pour la création des bases de données DB_{μ} et DB_{σ}, ainsi que pour les simulations avec les classificateurs *k*NN et quadratique Bayesien.

CHAPITRE 3

STRATÉGIES D'APPRENTISSAGE STANDARD ÉVALUÉES SUR LES BASES DE DONNÉES SYNTHÉTIQUES

Ce chapitre présente les résultats obtenus avec les bases de données synthétiques. Ces bases sont utilisées pour mesurer les performances des réseaux FAM au niveau des performances en généralisation, des temps de convergence et des taux de compression. De plus, les simulations ont permis de confirmer qu'il y a bien du sur-apprentissage dans les réseaux FAM.

Ce chapitre traite de plusieurs aspects considérés lors des simulations, soit: les effets dus à la taille de la base d'apprentissage, à la structure des bases de données, à la technique de normalisation, au degré de chevauchement ainsi qu'à la polarité du MatchTracking.

Tous ces aspects sont étudiés, cas par cas, pour bien mesurer leurs impacts sur les performances en généralisation, les temps de convergence ainsi que sur les taux de compression des réseaux FAM. De plus, lors des expériences, les paramètres internes du FAM sont les paramètres généraux soit: $\alpha = 0.01$, $\overline{\rho} = 0.0$, $\beta : 1.0$ et $\varepsilon = 0.001$ (MT+) ou -0.001 (MT-). Les résultats obtenus dans ce chapitre ont contribués à la publication d'un article [34].

3.1 Effets de la taille de la base d'apprentissage

Pour mesurer les effets engendrés par la taille de la base de données d'apprentissage, celle-ci est graduellement augmentée de 5 à 5000 patrons par classe en respectant une progression logarithmique (voir tableau II, section 2.1.1). L'étude de cet effet nous permet de vérifier la dégradation des performances en fonction de la taille de la base d'apprentissage.

De plus, l'utilisation de quatre stratégies d'apprentissage permet de montrer l'impact du nombre d'époques d'entraînement sur les performances des réseaux FAM et ainsi voir si ce facteur peut engendrer une dégradation des performances du réseau FAM.

Cette section est divisée en trois sous-sections. La première section présente les effets de la taille de la base d'apprentissage sur les bases de données synthétiques avec chevauchement et la deuxième section présente ces effets sur les bases de données synthétiques sans chevauchement. La troisième partie présente l'analyse des effets engendrés par la taille de la base d'apprentissage.

3.1.1 Bases de données avec chevauchement

Deux bases de données synthétiques avec chevauchement sont utilisées, soit DB_{μ} et DB_{σ}. La Figure 9 présente l'erreur en généralisation, le taux de compression ainsi que le temps de convergence obtenus avec la base $DB_{\mu}(1\%)$ pour les quatre stratégies d'apprentissage en utilisant la technique de normalisation MinMax. La Figure 10 et la Figure 11 présentent respectivement les mêmes types de résultats pour les bases $DB_{\mu}(9\%)$ et $DB_{\mu}(25\%)$. Les résultats obtenus par les autres degrés de chevauchement de la base DB_{μ} sont présentés dans l'annexe 3.

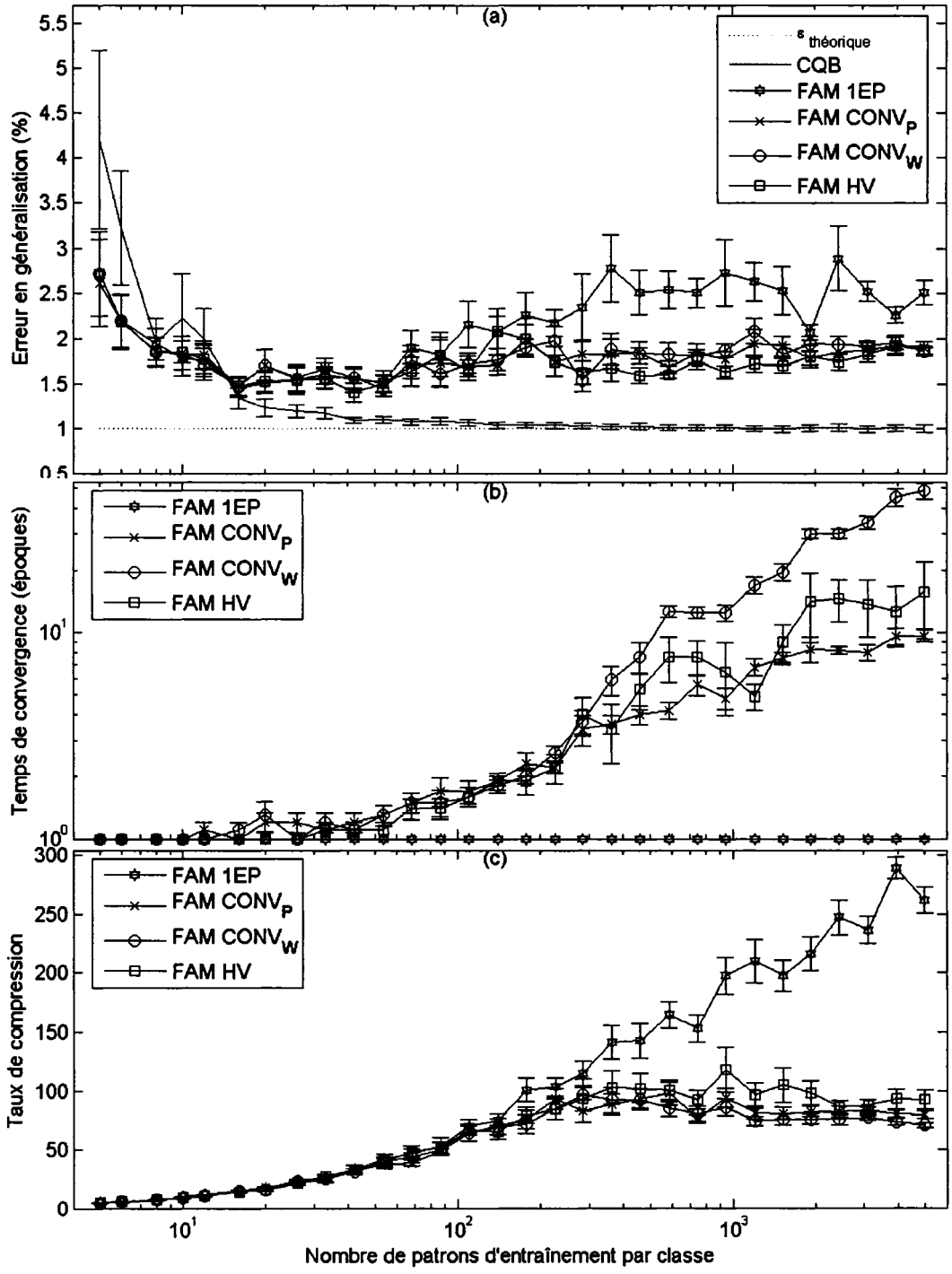


Figure 9 Performances moyennes du FAM en fonction de la taille de la base d'apprentissage avec la base $DB_\mu(1\%)$

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

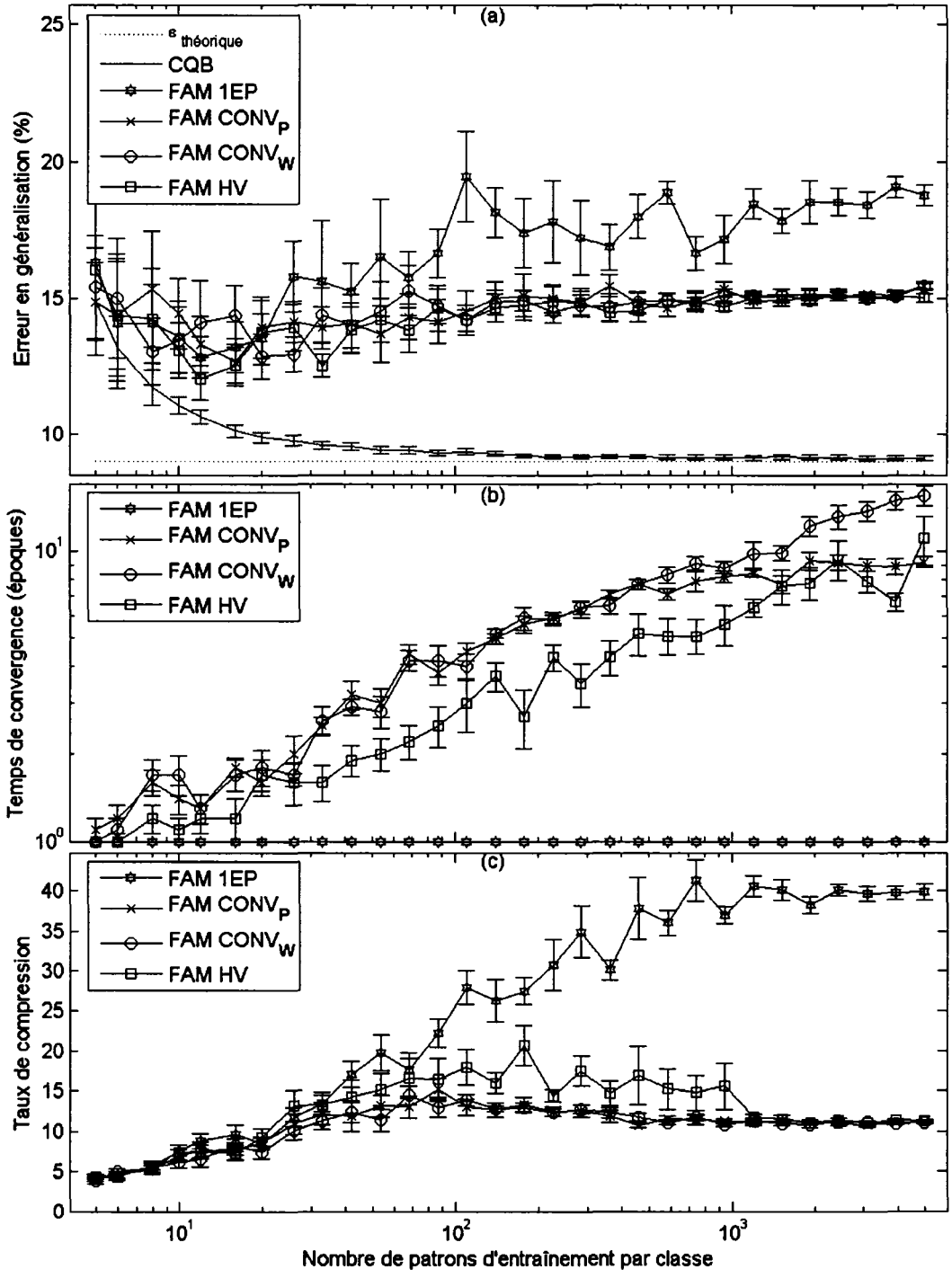


Figure 10 Performances moyennes du FAM en fonction de la taille de la base d'apprentissage avec la base $DB_{\mu}(9\%)$

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

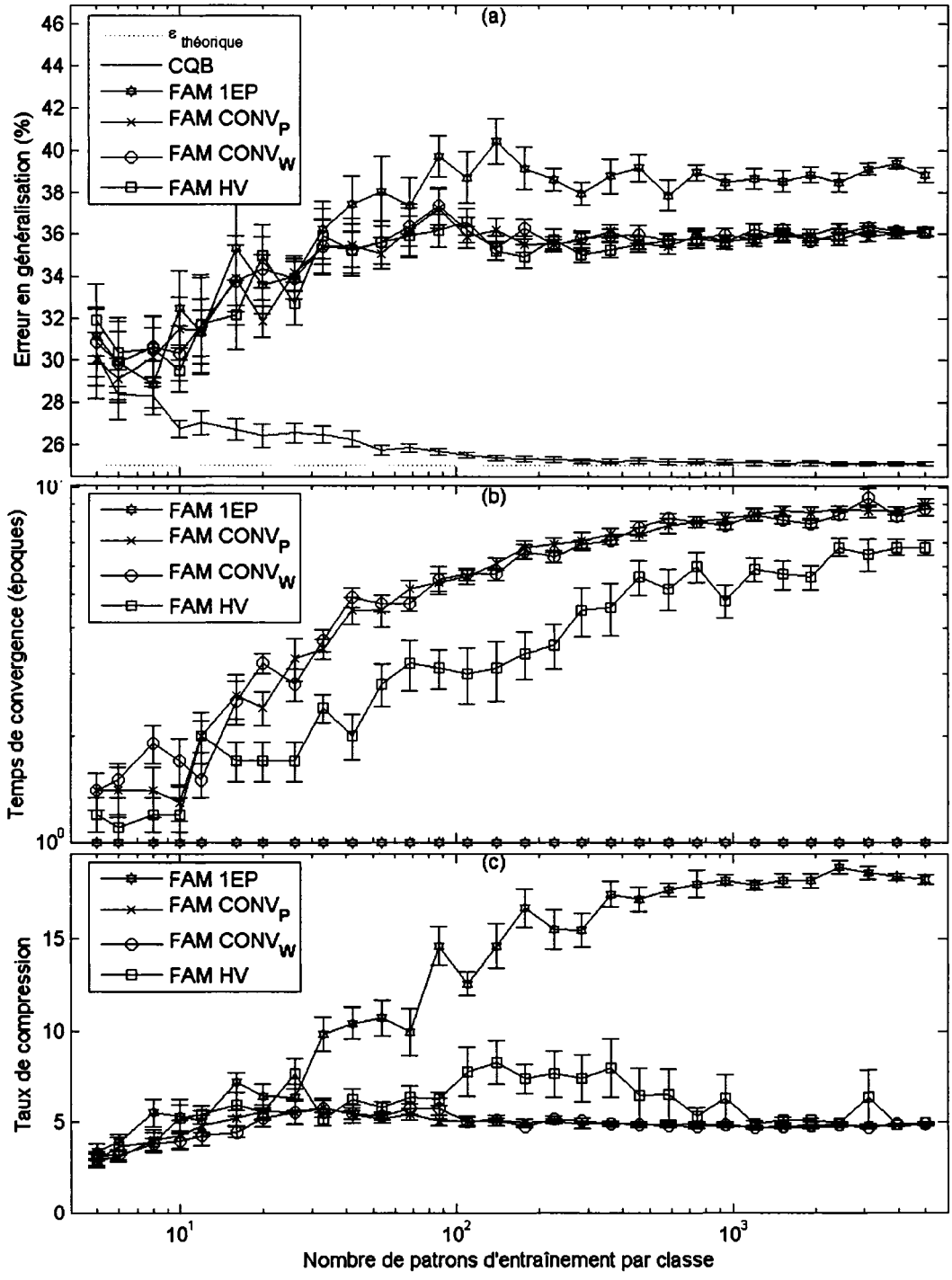


Figure 11 Performances moyennes du FAM en fonction de la taille de la base d'apprentissage avec la base $DB_\mu(25\%)$

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

3.1.2 Bases de données sans chevauchement

Deux bases de données sans chevauchement sont utilisées, soit DB_{CiS} et DB_{P2}. La Figure 12 présente les résultats généraux obtenus pour la base DB_{CiS}, soit l'erreur en généralisation, le temps de convergence ainsi que le taux de compression, pour les quatre stratégies d'apprentissage, lors de l'utilisation de la technique de normalisation MinMax. La Figure 13 présente les mêmes types de résultats généraux obtenus avec la base DB_{P2}.

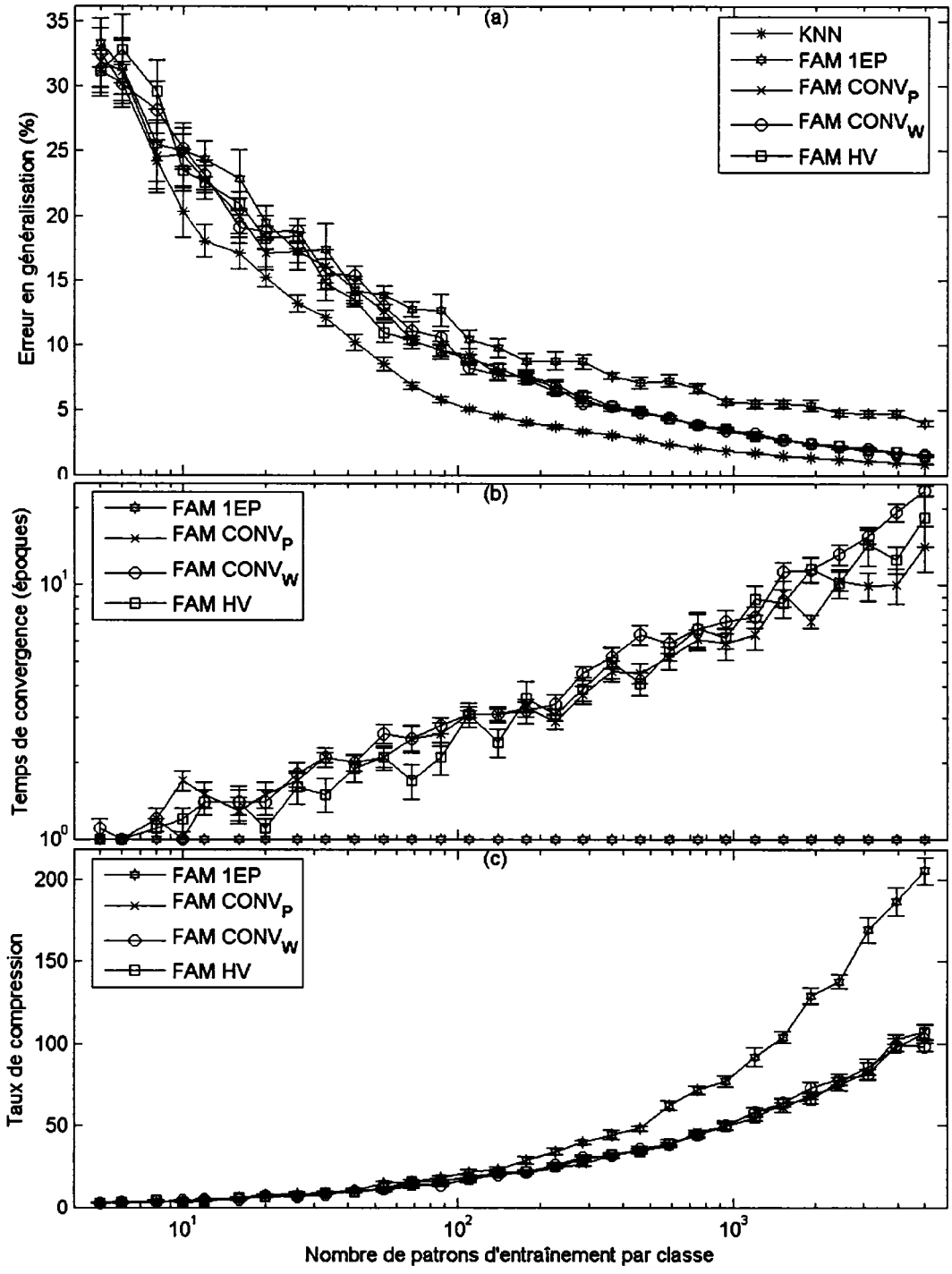


Figure 12 Performances moyennes du FAM en fonction de la taille de la base d'apprentissage avec la base DB_{CIS}

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

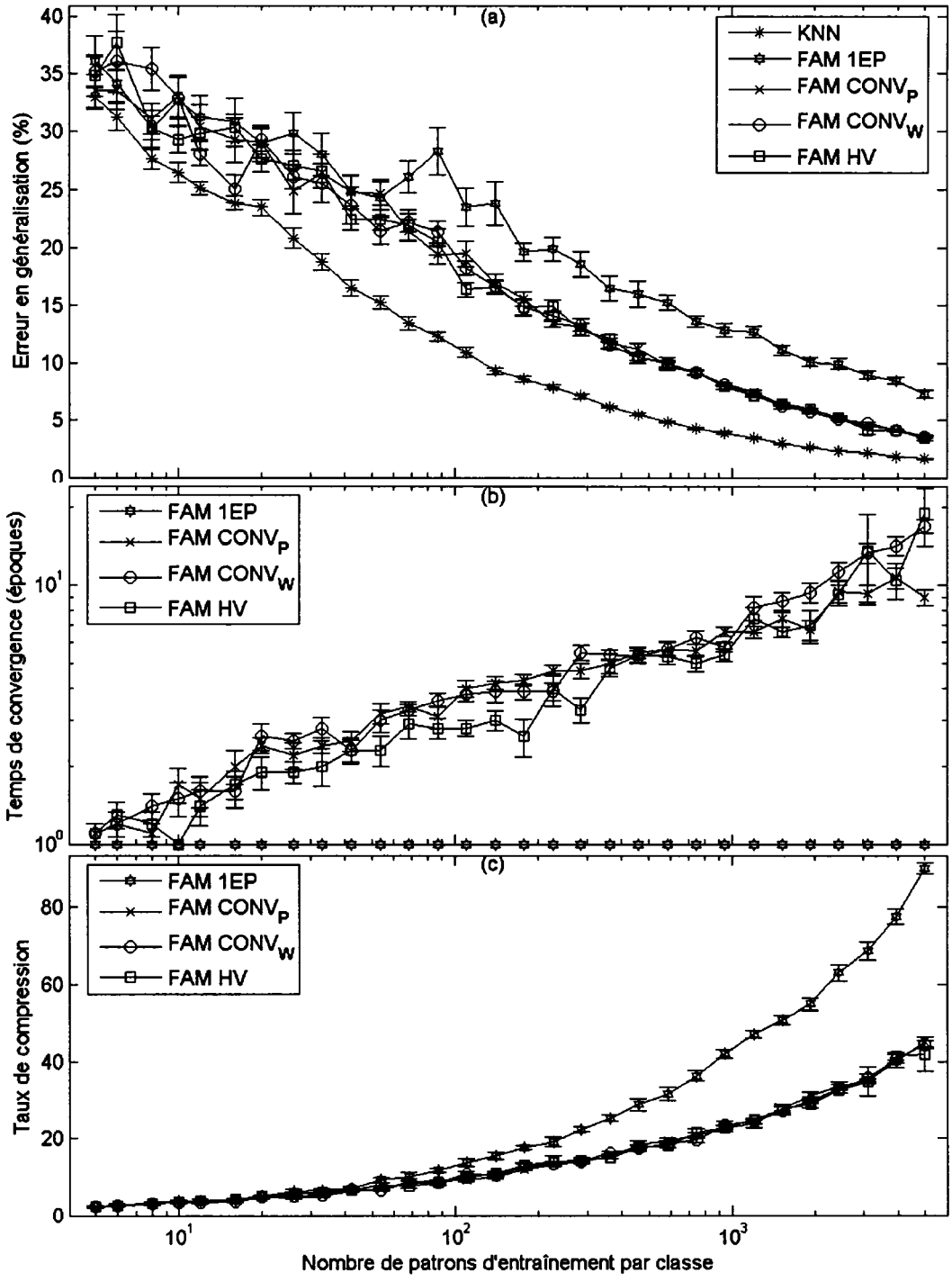


Figure 13 Performances moyennes du FAM en fonction du nombre de patrons d'entraînement avec la base DB_{P2}

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

3.1.3 Analyse

En observant les résultats obtenus avec les bases de données avec chevauchement, on constate que, peu importe la méthode d'apprentissage utilisée, l'erreur en généralisation des réseaux FAM augmente lorsque la taille de la base d'apprentissage dépasse un certain niveau. Ainsi, pour ces bases, il existe un effet de sur-apprentissage qui est fonction de la taille de la base d'entraînement.

La Figure 10(b) présente le taux de compression obtenu lors des tests avec la base de données $DB_{\mu}(9\%)$. On remarque que, lorsque les réseaux FAM commencent à subir une dégradation des performances en généralisation due à la taille de la base d'apprentissage, soit lorsque l'erreur en généralisation augmente, la compression des deux stratégies utilisant la convergence cesse d'augmenter.

Nous savons que la stratégie d'apprentissage HV est immunisée contre le phénomène de sur-apprentissage en fonction du nombre d'époques d'entraînement. Les stratégies $CONV_W$ et $CONV_P$ obtiennent des erreurs en généralisation légèrement supérieures ainsi que des temps de convergence supérieurs à la stratégie HV. En tenant compte de ces deux aspects, nous pouvons conclure que les réseaux fuzzy ARTMAP peuvent engendrer du sur-apprentissage en fonction du nombre d'époques d'entraînement. Cependant, plus la taille de la base d'apprentissage est grande, moins ce phénomène est apparent.

En regardant l'ensemble des résultats obtenus, en faisant varier la taille de la base d'apprentissage, on peut observer le phénomène suivant: dès qu'il y a chevauchement entre deux classes, il peut y avoir un effet de sur-apprentissage dû à la taille de la base d'apprentissage et au nombre d'époques d'apprentissage.

Analysons maintenant la manifestation de ce phénomène à l'intérieur des réseaux fuzzy ARTMAP. Lors de la phase d'apprentissage, la règle générale veut que le maximum de données soient utilisées. Cette règle s'appuie sur le concept que, plus il y a d'éléments dans la base d'apprentissage, meilleure est la représentation de la dispersion réelle des données. Cependant, les résultats obtenus démontrent que, dans certains cas, la performance des réseaux fuzzy ARTMAP chute lorsqu'il y a trop de données dans la base d'apprentissage. Ceci est dû à la prolifération des catégories, c'est-à-dire que le réseau final a créé trop de catégories lors de l'entraînement. Ce surplus de catégories diminue les performances qu'aurait pu atteindre le réseau et provient du fait que trop de patrons ont été présentés lors de la phase d'apprentissage.

La Figure 14(a) présente les bornes de décision (a.1) et les catégories (a.2) obtenues lors de l'entraînement avec la taille maximale de la base d'apprentissage (5000 patrons par classe) pour un des tests avec la base $DB_{\mu}(9\%)$ en utilisant la stratégie d'apprentissage HV.

Ce réseau FAM est composé de 916 catégories et obtient une erreur en généralisation de 14.85%, soit 5.85% de plus que l'erreur théorique fixe à 9%. Cette différence est due au grand nombre de petites zones en erreur, soit celles situées de l'autre côté de la frontière de décision optimale. Ces petites zones en erreur proviennent du fait que des catégories appartenant à la classe 1 sont totalement comprises dans une zone qui devrait statistiquement appartenir à la classe 2, et vice-versa. Il faut mentionner que les régions [(0,0.3);(0.7,1)] et [(0.7,1);(0,0.3)] n'influencent pratiquement pas la performance en généralisation, car un très faible nombre de données peuvent être situées dans ces deux régions.

La figure 14(b) présente les bornes de décision (b.1) et les catégories (b.2) obtenues avec 26 patrons d'apprentissage pour la même base de données et la même stratégie d'entraînement. Étant donné le faible nombre de patrons d'apprentissage utilisés, il n'y a

que 4 catégories créées lors de l'entraînement. Ce réseau obtient une erreur en généralisation plus faible qu'avec la taille maximum de la base d'apprentissage, soit de 9.87%, seulement 0.87% de plus que l'erreur théorique. Ceci crée une borne de décision beaucoup plus proche de la borne optimale et empêche la création des nombreuses petites zones en erreur tel que représenté par la Figure 14(a.1).

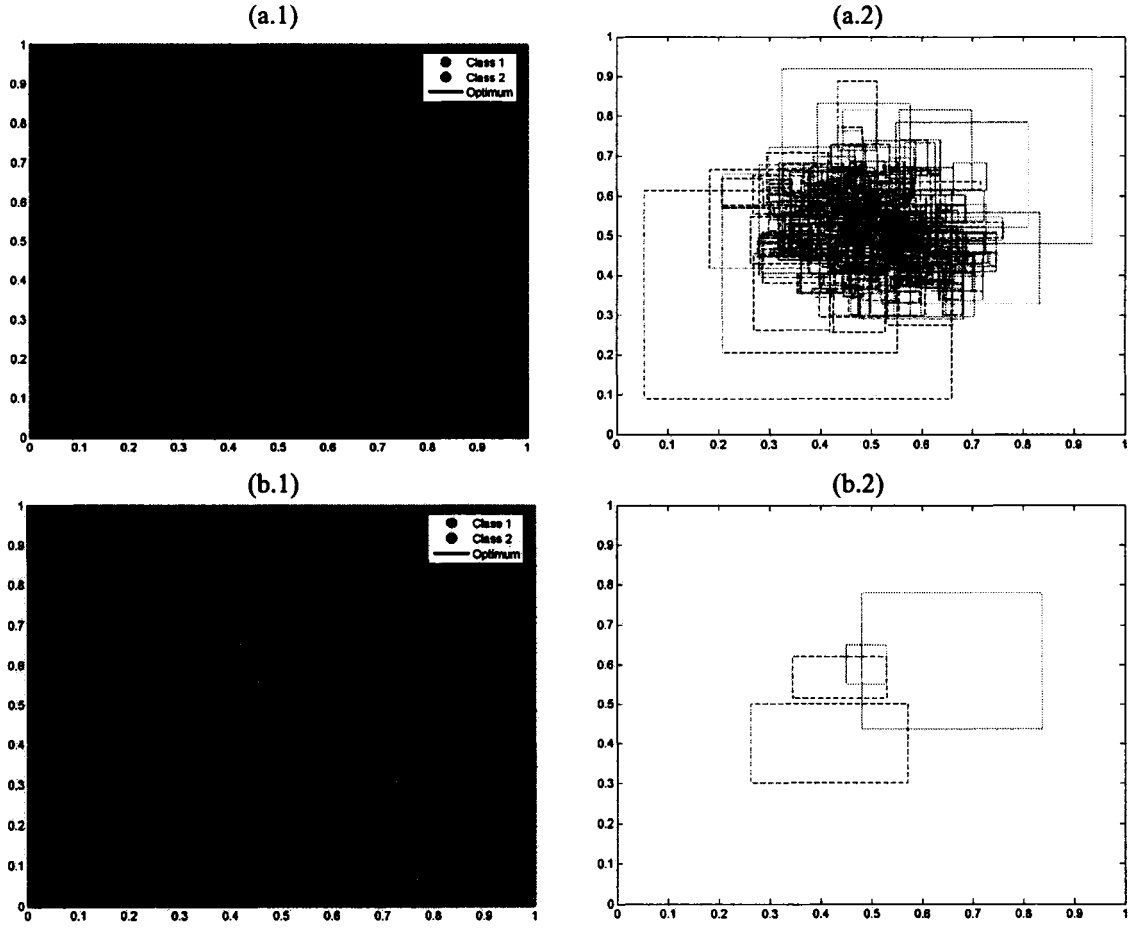


Figure 14 Catégories et bornes de décision obtenues pour $DB_{\mu}(9\%)$ avec HV Soit (a) 5000 patrons et (b) 26 patrons d'entraînement par classe, et (.1) bornes de décision et (.2) catégories créées.

Ainsi, en optimisant la taille de la base d'apprentissage, sur un des tests avec la base $DB_{\mu}(9\%)$, nous avons réussi à passer d'une erreur en généralisation de 14.85% à 9.87%,

soit une réduction de 4.98%. Il y a donc un net avantage à optimiser la taille de la base d'entraînement pour les bases de données possédant un degré de chevauchement.

En observant les résultats obtenus avec les bases de données sans chevauchement (DB_{CiS} et DB_{P2}), il est évident qu'il n'y a aucune dégradation des performances du FAM en fonction de la taille de la base d'apprentissage. Ceci provient du fait qu'il n'y a pas de chevauchement entre les deux classes. Les nombreuses petites zones que nous avons observées avec les bases de données possédant un degré de chevauchement (voir figure 14) sont inexistantes car toutes les catégories créées sont totalement ou en partie dans leur zone optimale.

La Figure 15 présente les bornes de décision ainsi que les catégories créées avec la base de données DB_{CiS} lors de l'augmentation progressive du nombre de patrons d'entraînement, soit 10, 178 et 5000 patrons par classe. La stratégie d'apprentissage HV ainsi que la technique de normalisation MinMax sont utilisées. On constate que, plus la taille de la base d'apprentissage augmente, plus le nombre de catégories dans la région de transition entre les deux classes augmente, créant ainsi une borne de décision de plus en plus proche de la borne optimale. Cette augmentation de la base de données fait passer l'erreur en généralisation de 30.00% (10 patrons par classe), à 7.64% (178 patrons par classe) et finalement à 1.66% (5000 patrons par classe). On observe également le même phénomène avec la base de donnée DB_{P2}. Ainsi, la prolifération des catégories est une bonne chose dans le cas des distributions sans chevauchement car elle raffine la borne de décision pour ce type de problèmes.

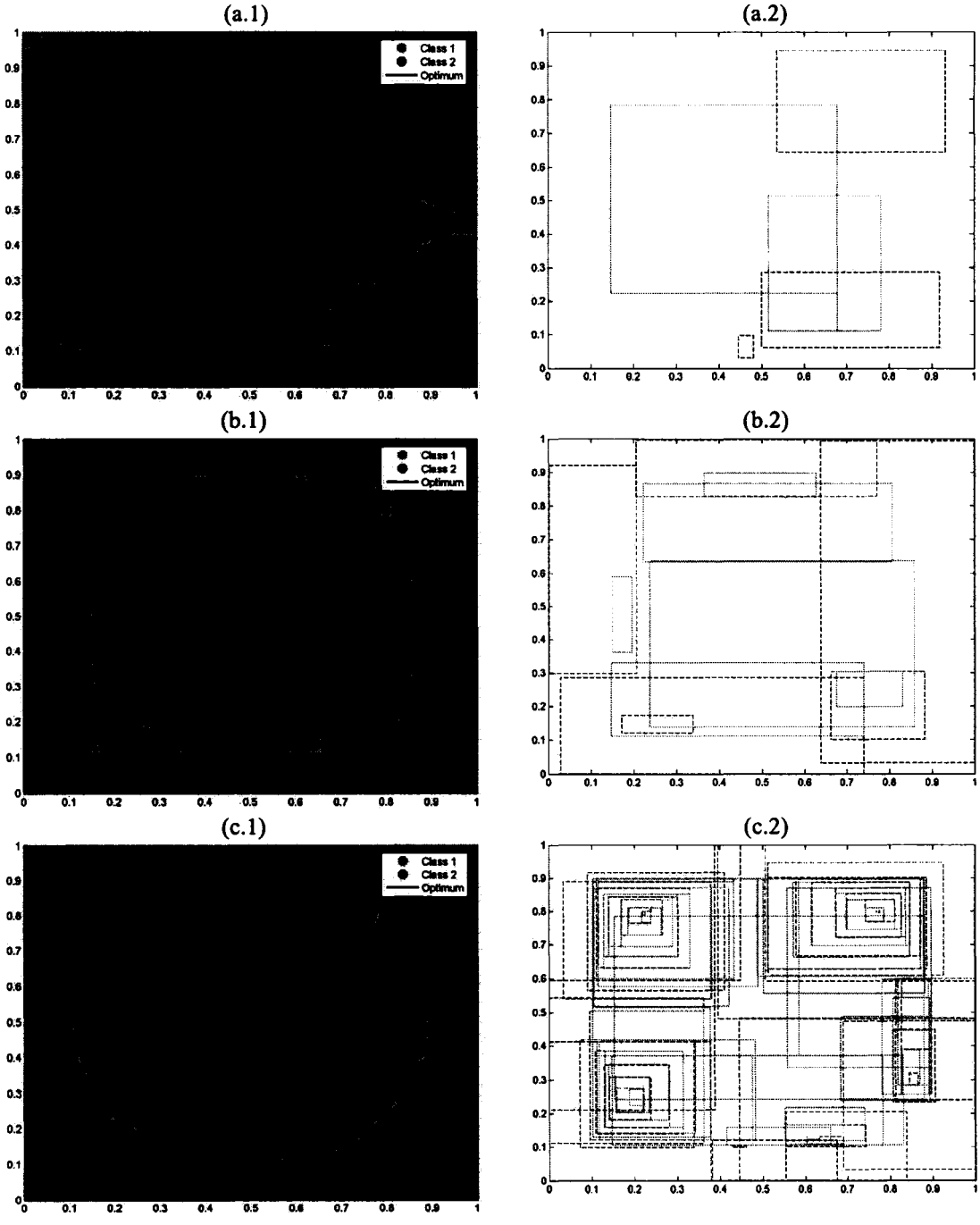


Figure 15 Bornes et catégories obtenues lors de l'accroissement de la taille de la base d'apprentissage avec la base $DB_{CIS,}$.

Soit (a) 10 patrons, (b) 178 patrons et (c) 5000 patrons d'entraînement par classe, et (.1) bornes de décision et (.2) catégories créées.

De plus, aucune dégradation des performances du réseau FAM en fonction du nombre d'époques d'entraînement n'est visible dans les résultats des bases DB_{CiS} et DB_{P2}. En effet, bien que les temps de convergence entre les stratégies d'apprentissage ne soient pas identiques, il n'y a pas de différences significatives entre les performances en généralisation et les taux de compression obtenus par les diverses stratégies d'apprentissage, mis à part l'entraînement sur une époque (1EP).

Bref, peu importe qu'il y ait ou non du chevauchement entre les classes, la stratégie d'apprentissage 1EP génère, en moyenne, le taux de compression le plus élevé ainsi que la plus grande erreur en généralisation. La stratégie d'apprentissage HV donne, en moyenne, des résultats meilleurs ou équivalents aux deux stratégies basées sur la convergence des poids synaptiques et des patrons.

Pour mesurer les effets dus au degré de chevauchement entre les classes, celui-ci est progressivement augmenté de 1% à 25% (voir tableau III et Tableau IV, section 2.1.1) pour les deux bases DB_{μ} et DB_{σ}. Puisque nous venons de démontrer que le sur-apprentissage en fonction de la taille de la base d'entraînement peut exister dans les réseaux fuzzy ARTMAP, pour les bases de données avec chevauchement, nous pouvons présenter son évolution en fonction du degré de chevauchement. Pour ce faire, nous allons analyser l'impact du degré de chevauchement avec les quatre stratégies d'apprentissage.

Définissons l'erreur de sur-apprentissage comme étant la différence entre l'erreur en généralisation obtenue avec la grandeur maximale de la base d'apprentissage (5000 patrons par classe) et l'erreur en généralisation minimale obtenue sur l'ensemble de la taille de la base d'apprentissage. En appliquant l'équation (3.1), nous obtenons la moyenne de l'erreur de sur-apprentissage E_{sapp} évaluée sur l'ensemble des simulations que nous avons effectuées.

$$E_{sapp} = \frac{\sum_{i=1}^{N} E_{gen}(5k/\omega)_i - E_{gen}(\min)_i}{N} \tag{3.1}$$

Où : N est le nombre de répétitions lors des expérimentations *(10)*

$E_{gen}(5k/\omega)_i$ est l'erreur en généralisation obtenue avec 5000 patrons d'entraînement par classe, soit la taille maximale de la base d'apprentissage, de la $i^{ème}$ répétition.

$E_{gen}(\min)_i$ est l'erreur en généralisation minimale obtenue sur l'ensemble des tailles de la base d'apprentissage de la $i^{ème}$ répétition.

La Figure 16 présente la moyenne des erreurs de sur-apprentissage obtenue lors des tests avec les diverses bases de données DB_μ.

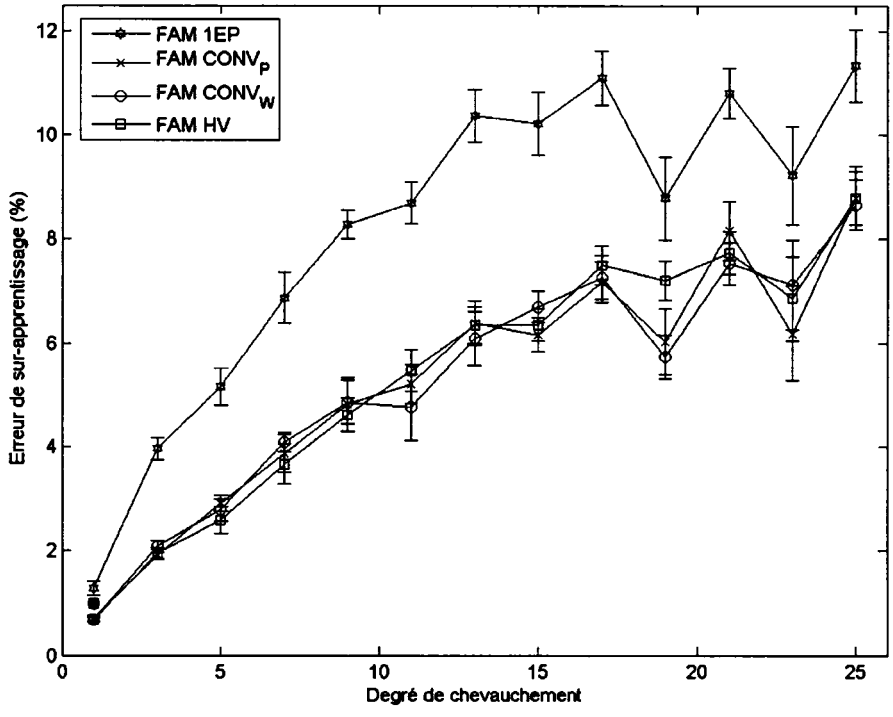


Figure 16 Erreurs de sur-apprentissage dues à la taille de la base d'apprentissage en fonction du degré de chevauchement

On remarque que plus le degré de chevauchement est grand, plus la taille de la base d'apprentissage joue un rôle important dans la dégradation des performances. De plus, nous avons constaté que plus le degré de chevauchement est grand, plus la taille optimale de la base d'entraînement est petite et vice-versa. Ainsi, avec une base de données sans chevauchement, il faut utiliser le maximum de données d'entraînement pour obtenir les meilleures performances en généralisation.

Les stratégies d'apprentissage $CONV_W$, $CONV_P$ et la HV génèrent des erreurs de sur-apprentissage semblables. Ainsi, elles obtiennent le même type d'écart entre les erreurs en généralisation minimales obtenues lors des répétitions, et les erreurs en généralisations obtenues avec la taille maximale de la base d'entraînement.

En regardant les résultats de la Figure 10, on pourrait conclure que l'erreur de sur-apprentissage pour la base $DB_\mu(9\%)$, lors de l'utilisation de la stratégie HV, ne devrait pas dépasser 3%. Pourtant, la Figure 16 montre qu'il y a environ 4.5% d'erreur de sur-apprentissage avec ce degré de chevauchement. Cet écart provient du fait que cette erreur est calculée individuellement sur chacune des 10 réplications et que la taille optimale de chacune, bien que située dans la même région, n'est pas la même. Ainsi, même si l'on peut faire une approximation de l'erreur de sur-apprentissage avec la Figure 10, elle ne peut pas servir à établir convenablement cette erreur car elle représente la moyenne obtenue sur chaque taille et non les courbes individuelles des 10 réplications.

Définissons également l'erreur nette comme étant la différence entre l'erreur en généralisation obtenue pour chaque répétition et l'erreur théorique de la base de données. Cette valeur indique l'éloignement du réseau FAM face à l'erreur théorique.

La Figure 17 présente l'erreur nette obtenue par chaque stratégie d'apprentissage lors de l'optimisation de la taille de la base d'entraînement pour chaque réplication. On

remarque que plus le degré de chevauchement est élevé, plus l'erreur nette est grande. De plus, les quatre stratégies d'apprentissage obtiennent des erreurs nettes semblables pour un degré de chevauchement donné.

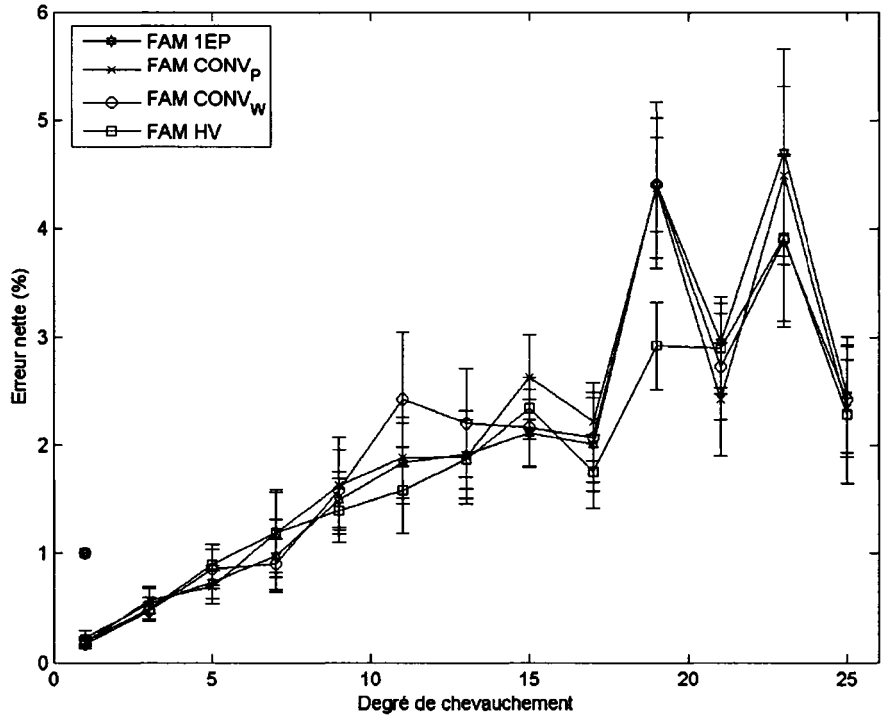


Figure 17 Erreur nette obtenue en sélectionnant la meilleur taille de la base d'apprentissage en fonction du degré de chevauchement

Pour démontrer à quel point il est important d'optimiser la taille de la base d'apprentissage pour les bases de données possédant un degré de chevauchement, la Figure 18 présente une comparaison entre l'erreur nette obtenue lors de l'optimisation de la taille de la base d'apprentissage et celle obtenue avec la taille maximale de la base d'apprentissage, soit 5000 patrons par classe; le tout pour la stratégie HV.

Cette figure démontre bien à quel point la taille de la base d'entraînement joue un rôle important dans la dégradation des performances des réseaux fuzzy ARTMAP lors de l'utilisation de bases avec chevauchement.

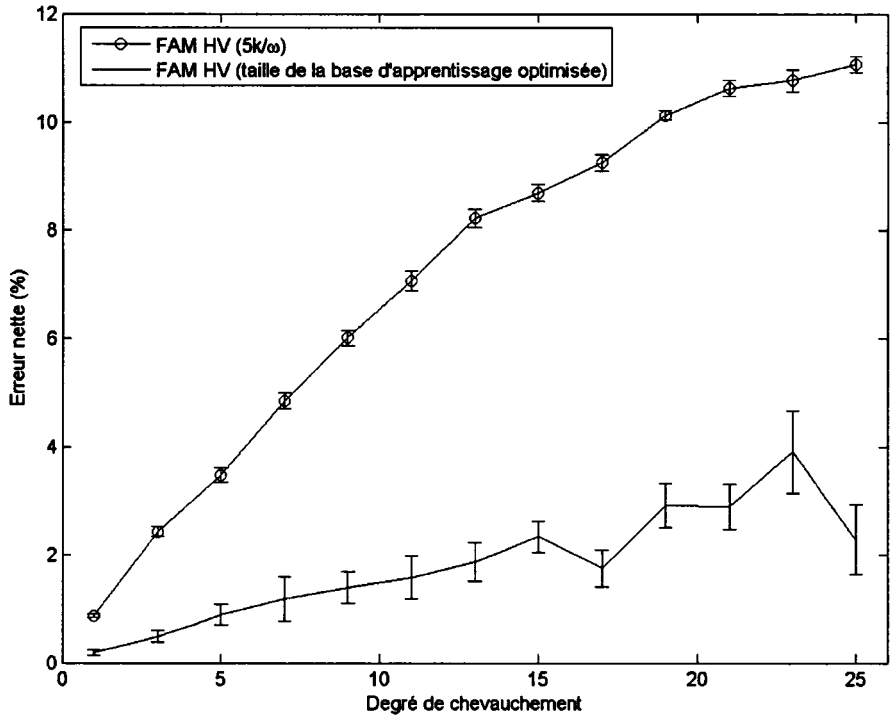


Figure 18 Erreur nette avec et sans optimisation du nombre de patrons d'apprentissage

3.2 Effets des structures des bases de données

Il existe plusieurs méthodes pour créer une base de données possédant un degré de chevauchement donné. Deux méthodes différentes ont été utilisées lors de la création des deux bases de données contenant du chevauchement. La première base, DB_{μ}, consiste en deux distributions gaussiennes dont la moyenne de la deuxième distribution est graduellement rapprochée de la moyenne de la première, augmentant ainsi le degré de chevauchement entre les distributions. La deuxième base, DB_{σ}, consiste en deux

distributions gaussiennes dont la variance des deux distributions est graduellement augmentée pour amplifier le degré de chevauchement entre les deux classes. Il résulte que, pour un taux d'erreur total identique, la frontière de décision entre les classes est plus grande avec DB_{σ}. Les paramètres utilisés lors de la création de ces deux bases de données sont présentés par le tableau III et le Tableau IV (section 2.1.1). Deux méthodes différentes ont également été utilisées lors de la création des deux bases de données sans chevauchement (DB_{CiS} et DB_{P2}).

3.2.1 Bases de données avec chevauchement

Cette section traite des différences obtenues entre les résultats des bases ayant un même degré de chevauchement mais dont la méthode de création diffère. Les figures 18 à 20 présentent ces différences pour une base de données de 9% de chevauchement. Les courbes présentes dans ces figures montrent les différences entre les résultats de DB_{μ} et de DB_{σ}. Ainsi, lorsque la courbe est positive, la valeur obtenue avec la base DB_{μ} est plus grande que celle obtenue avec la base DB_{σ}, et vice-versa. Ces figures présentent ces différences pour l'erreur en généralisation, le taux de compression ainsi que pour le temps de convergence. La technique de normalisation MinMax est utilisée. L'annexe 4 présente ces résultats pour les autres degrés de chevauchement utilisés.

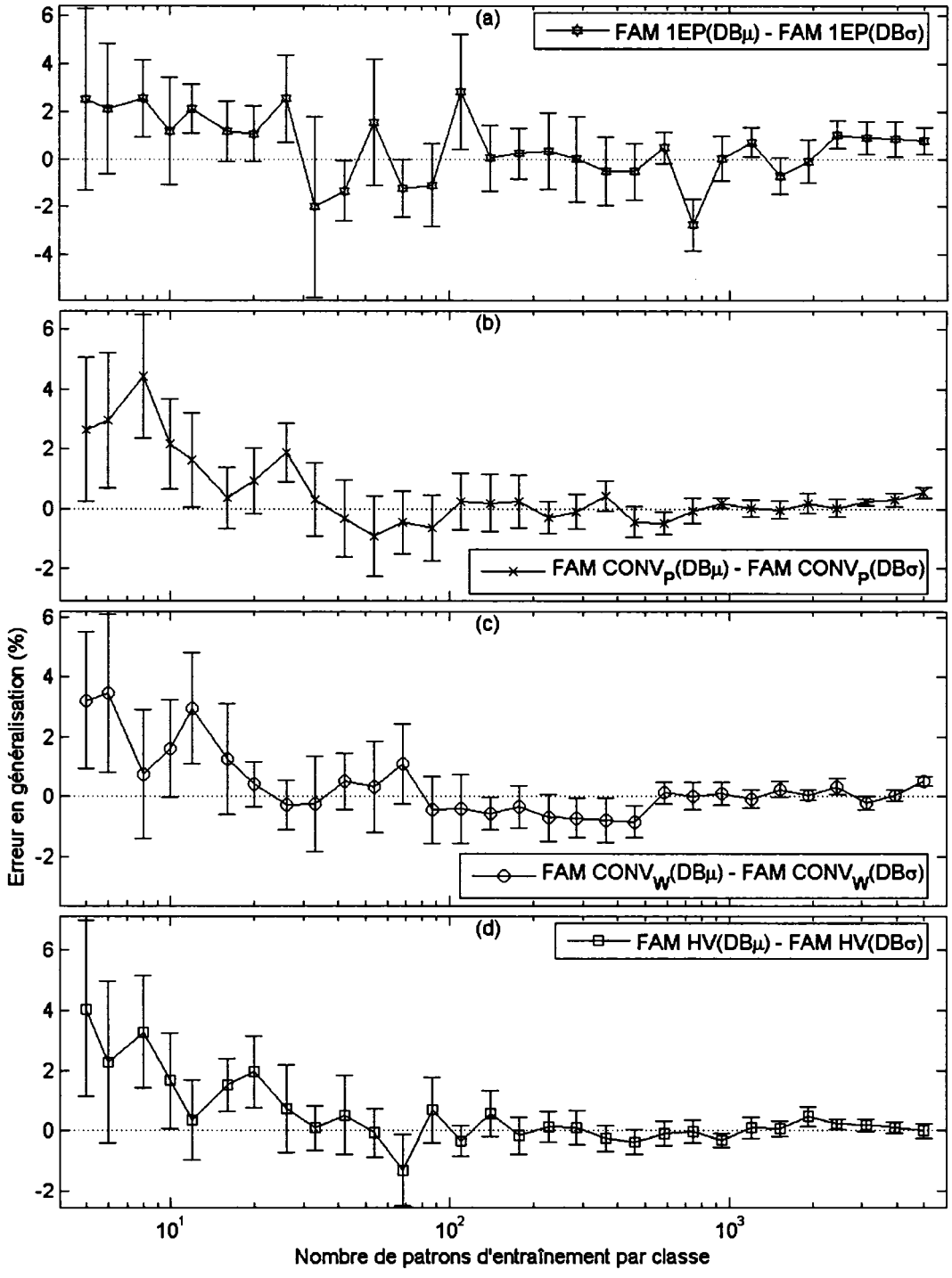


Figure 19 Différence entre $DB_\mu(9\%)$ et $DB_\sigma(9\%)$ sur l'erreur en généralisation (a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

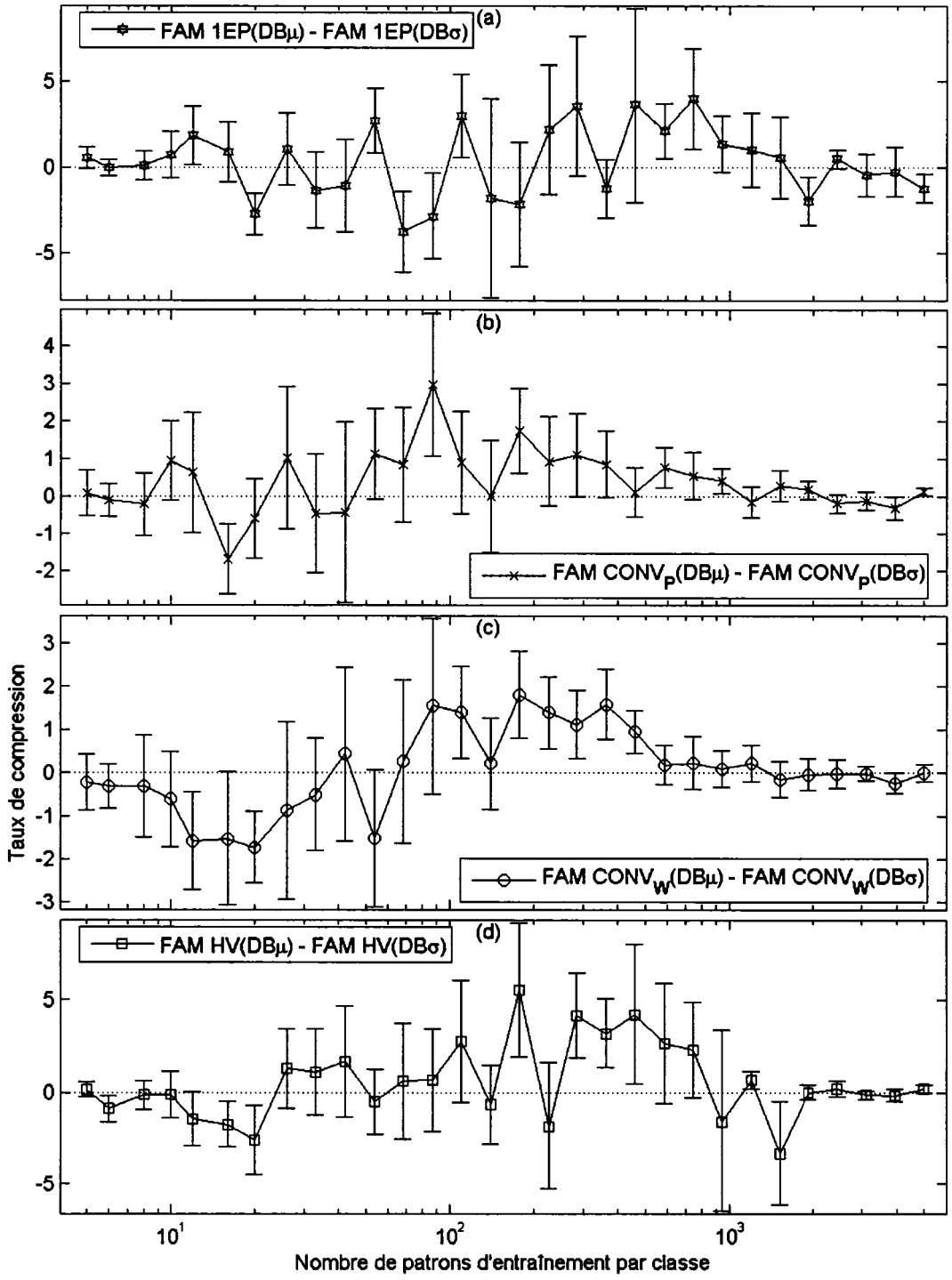


Figure 20 Différence entre $DB_\mu(9\%)$ et $DB_\sigma(9\%)$ sur le taux de compression (a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

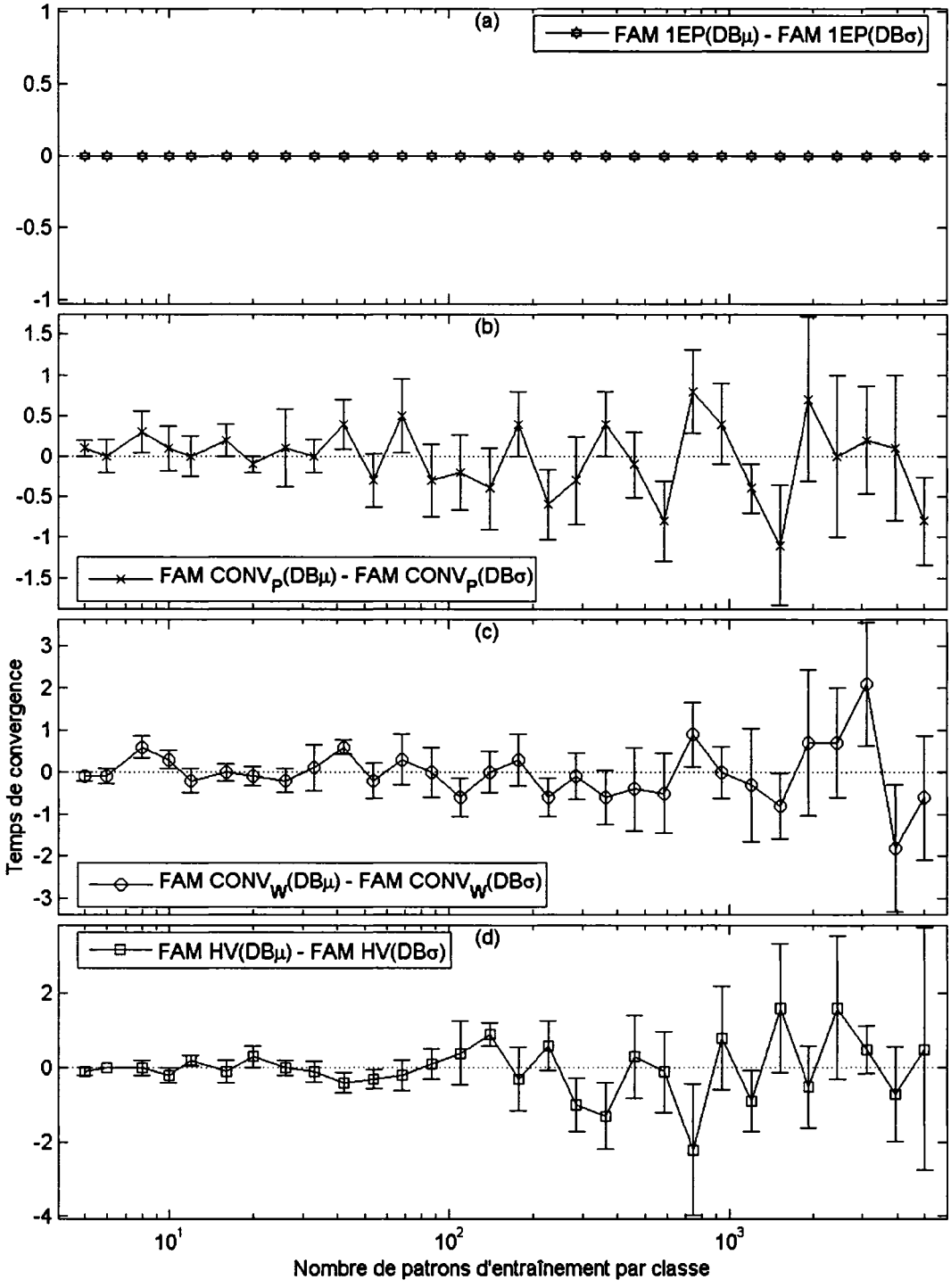


Figure 21 Différence entre $DB_{\mu}(9\%)$ et $DB_{\sigma}(9\%)$ sur le temps de convergence (a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

3.2.2 Bases de données sans chevauchement

Bien que les deux bases de données sans chevauchement utilisées aient la même erreur théorique, elles ne représentent pas pour autant le même niveau de difficulté. En effet, la base DB_{P2} possède plusieurs frontières de décision d'un degré de complexité élevé comparativement à l'unique frontière de décision de la base DB_{CiS}.

Les figures 21 à 23 présentent une comparaison entre les deux problèmes DB_{P2} et DB_{CiS} pour les quatre stratégies d'apprentissage en fonction de la taille de la base d'apprentissage. La normalisation MinMax est utilisée.

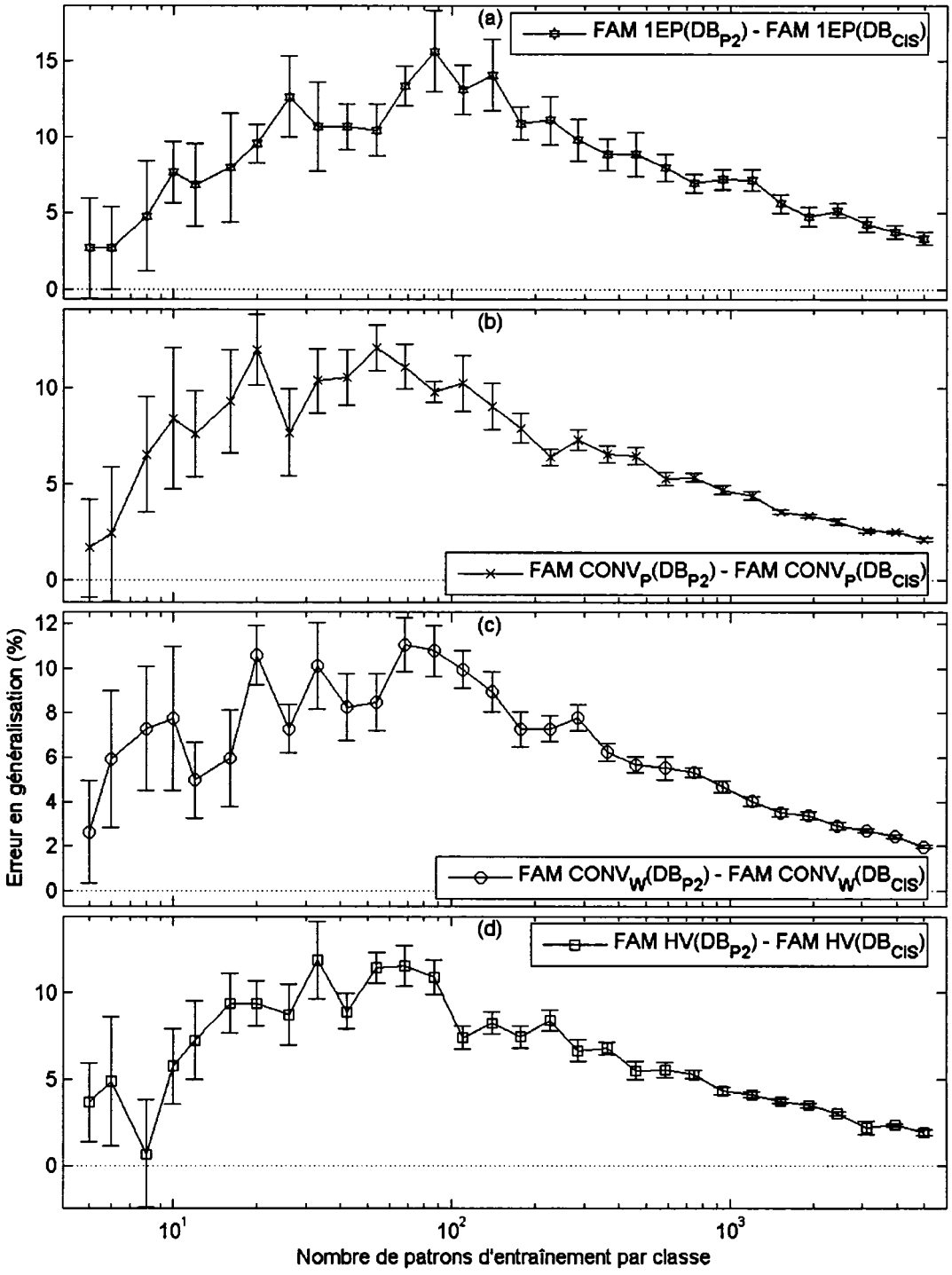


Figure 22 Différence entre DB_{CIS} et DB_{P2} sur l'erreur en généralisation

(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

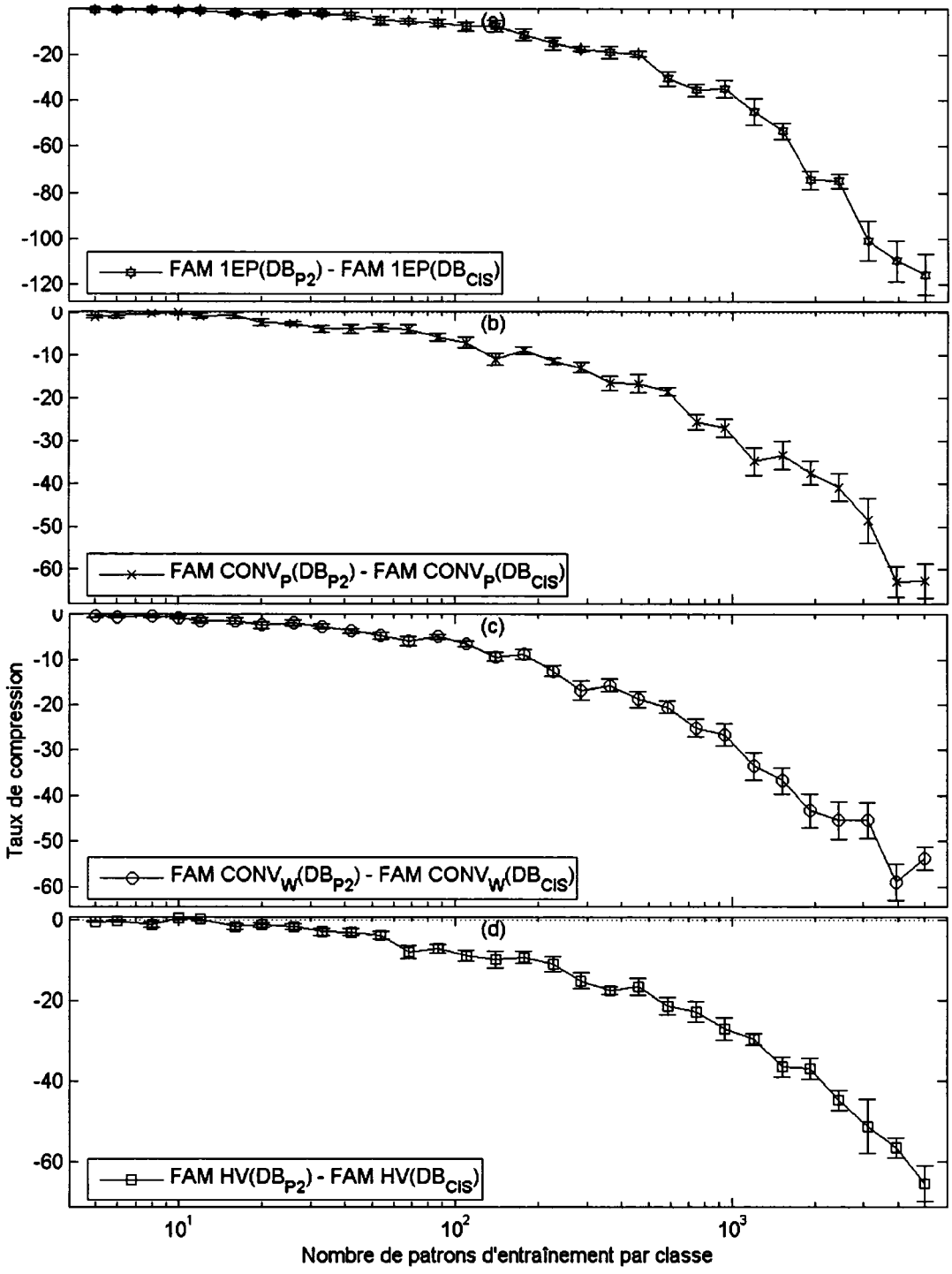


Figure 23 Différence entre DB_{CIS} et DB_{P2} sur le taux de compression

(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

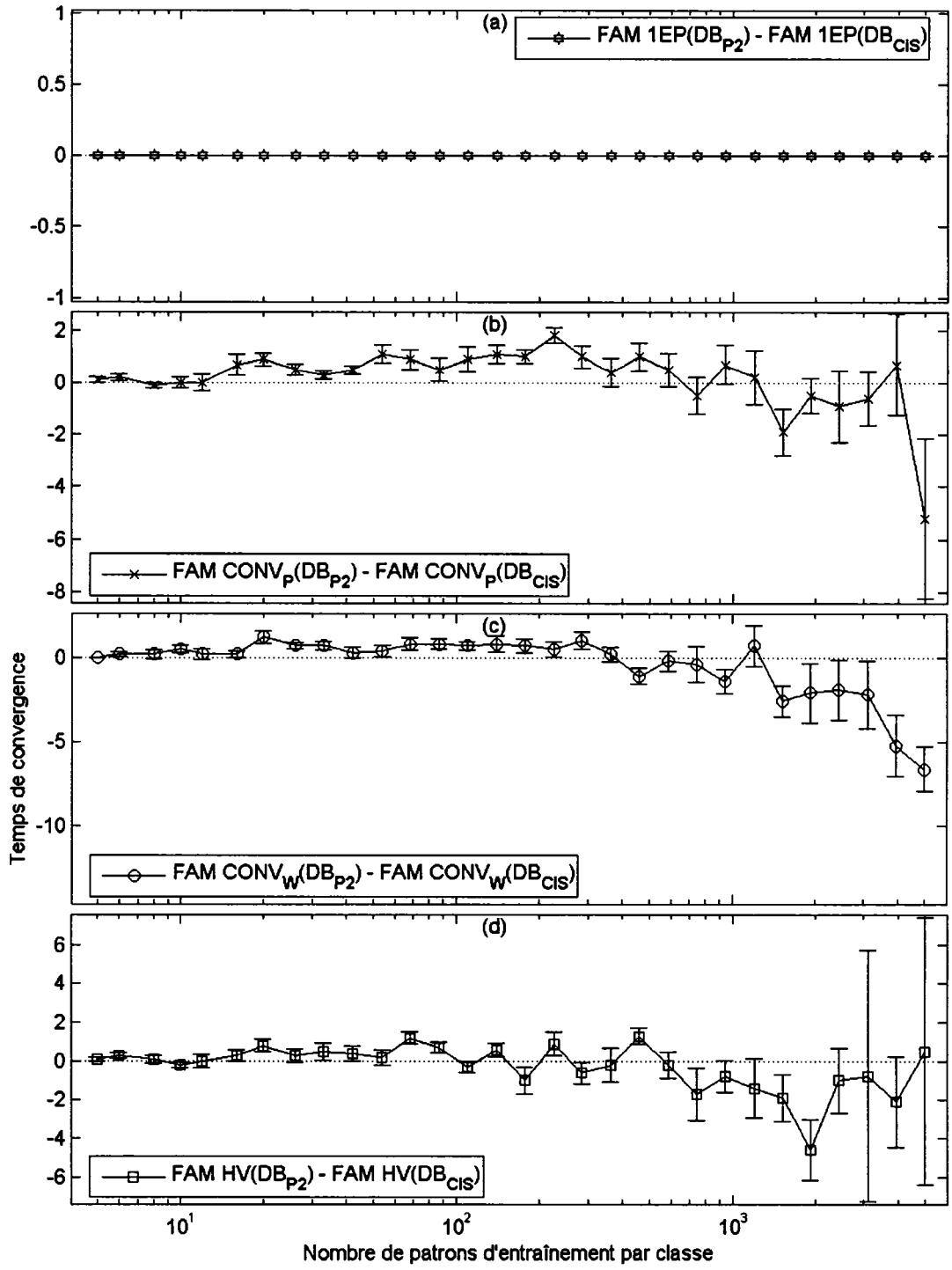


Figure 24 Différence entre DB_{CIS} et DB_{P2} sur le temps de convergence

(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

3.2.3 Analyse

En observant les résultats obtenus avec les bases de données avec chevauchement, on ne constate aucune différence majeure entre les deux méthodes de création utilisées. Les erreurs en généralisation, les temps de convergence ainsi que les niveaux de compression sont semblables entre ces deux bases pour un même degré de chevauchement. Aucune des deux structures des bases de données avec chevauchement testées ne montrent un avantage face à l'autre avec les réseaux fuzzy ARTMAP.

En observant les résultats obtenus avec les bases de données sans degré de chevauchement, on constate que les deux bases de données réagissent de la même manière. Ces deux bases ne montrent aucune dégradation des performances due à la taille de la base d'apprentissage et au nombre d'époques. Bien que les courbes présentant les erreurs en généralisation, les temps de convergence ainsi que les niveaux de compression aient les mêmes tendances, il y a de nettes différences au niveau de l'erreur en généralisation et du taux de compression.

Cet effet provient de la complexité des frontières de décision qui jouent un rôle important dans la performance des réseaux fuzzy ARTMAP. Les frontières de décision de la base DB_{P2} sont plus complexes que celles de la base DB_{CiS}. Cette différence engendre pour la base DB_{P2} des erreurs en généralisation plus élevées ainsi que des niveaux de compression plus faibles que ceux obtenus avec la base DB_{CiS}, et ce, pour une même taille de base d'apprentissage.

3.3 Effets de la normalisation

Tel que décrit dans le chapitre 2, deux méthodes de normalisation sont utilisées lors de la réalisation des expérimentations. Ces deux méthodes permettent de comprendre l'impact engendré par la normalisation des données. Les deux méthodes de

normalisation testées sont la normalisation MinMax ainsi que la normalisation Centrée Réduite décrites à la section 2.4.

En étudiant l'impact de ces deux méthodes de normalisation sur les performances en généralisation ainsi que sur la complexité des réseaux créés par le fuzzy ARTMAP nous pourrons favoriser l'une ou l'autre technique de normalisation face à un type de problème de classification donné.

3.3.1 Bases de données avec chevauchement

Cette section présente les différences obtenues entre les résultats des bases ayant un même degré de chevauchement mais dont la technique de normalisation diffère. Les figures 24 à 26 présentent une comparaison entre les techniques de normalisation pour les quatre stratégies d'apprentissage en fonction de la taille de la base d'apprentissage. Étant donné qu'aucune différence n'a été détectée entre les bases DB_{μ} et DB_{σ}, la base DB_{μ} est utilisée.

Le degré de chevauchement de la base DB_{μ} utilisé est de 9%. L'annexe 5 présente ces résultats pour tous les autres degrés de chevauchement.

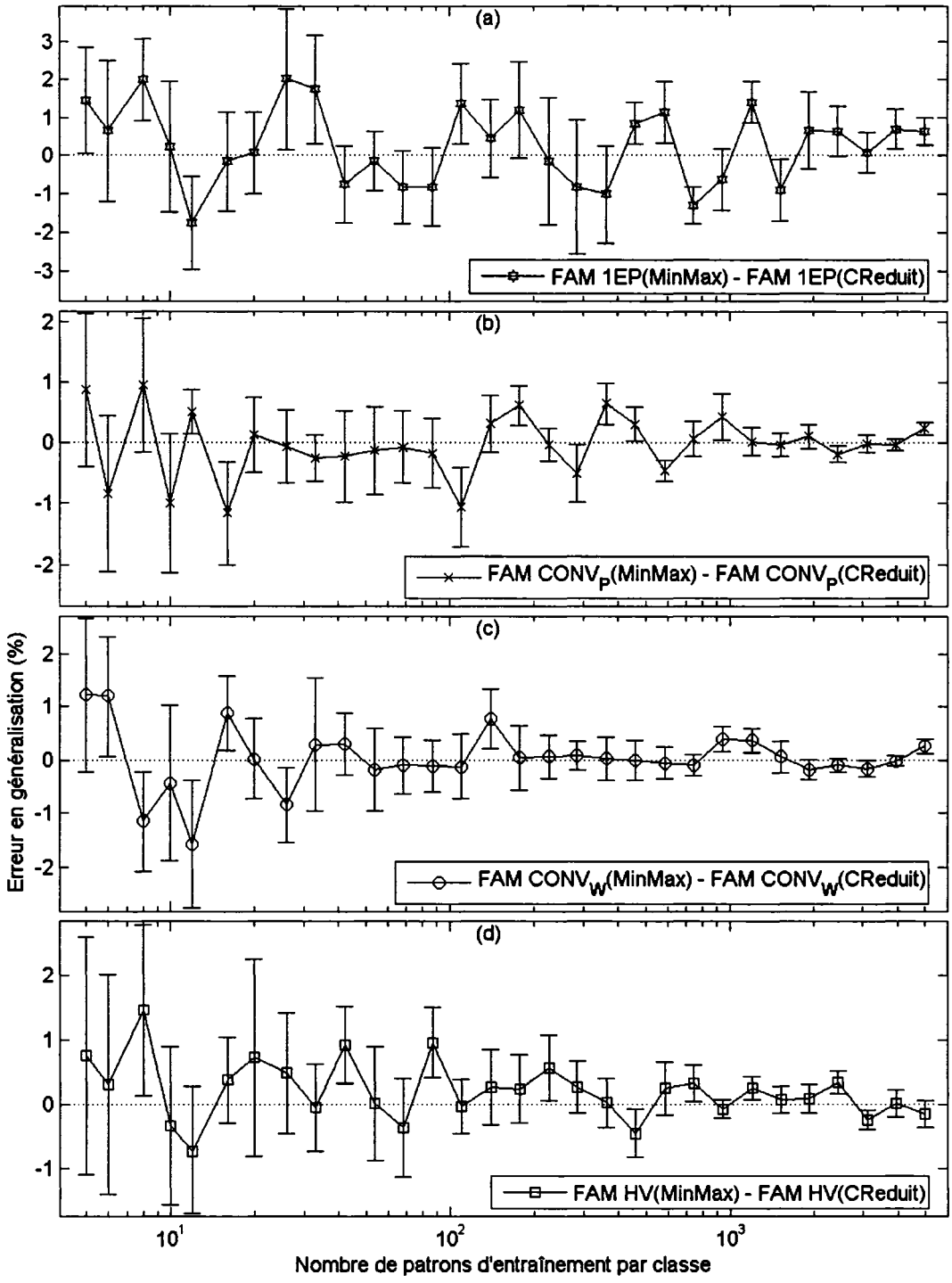


Figure 25 Effet de la normalisation sur l'erreur en généralisation avec $DB_\mu(9\%)$
(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

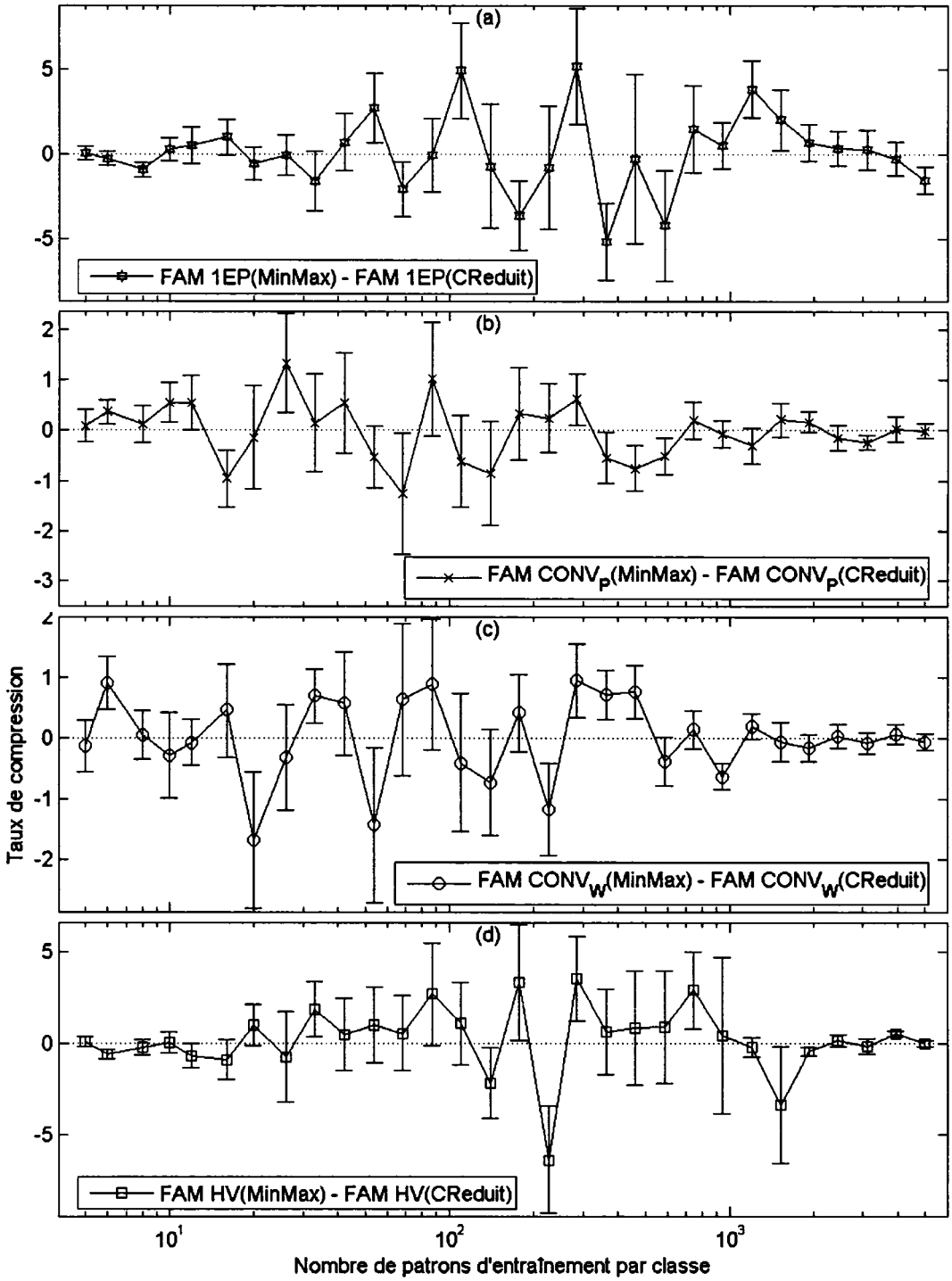


Figure 26 Effet de la normalisation sur le taux de compression avec $DB_\mu(9\%)$

(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out .

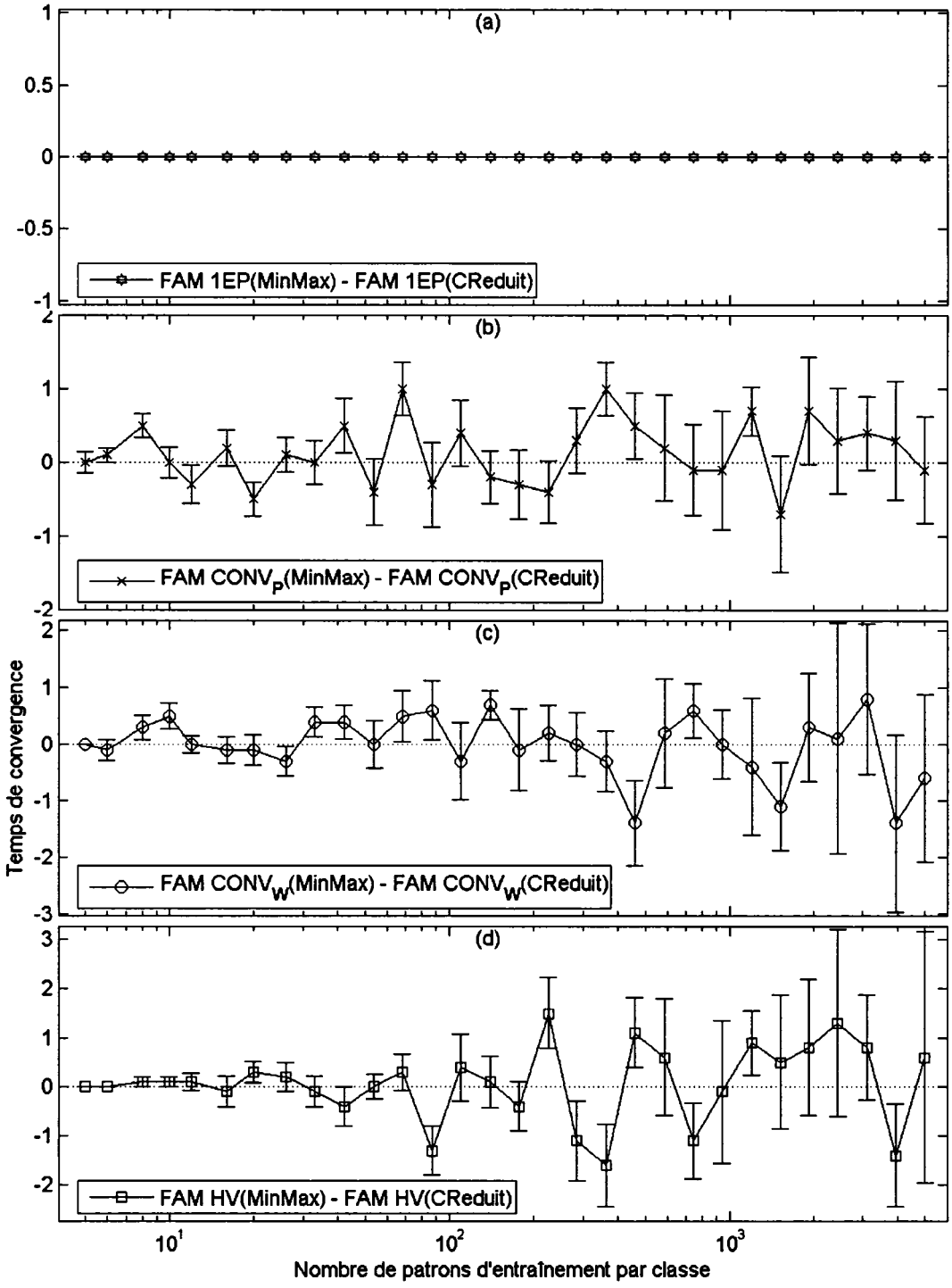


Figure 27 Effet de la normalisation sur le temps de convergence avec $DB_\mu(9\%)$
(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out

3.3.2 Bases de données sans chevauchement

Les figures 27 à 29 présentent une comparaison entre les deux méthodes de normalisation pour les quatre stratégies d'apprentissage en fonction de la taille de la base d'apprentissage pour la base de données sans chevauchement DB_{CiS}. Cette comparaison présente les erreurs en généralisation, les temps de convergence ainsi que les niveaux de compression. Les résultats obtenus avec la base DB_{P2} sont présentés à la fin de l'annexe 5.

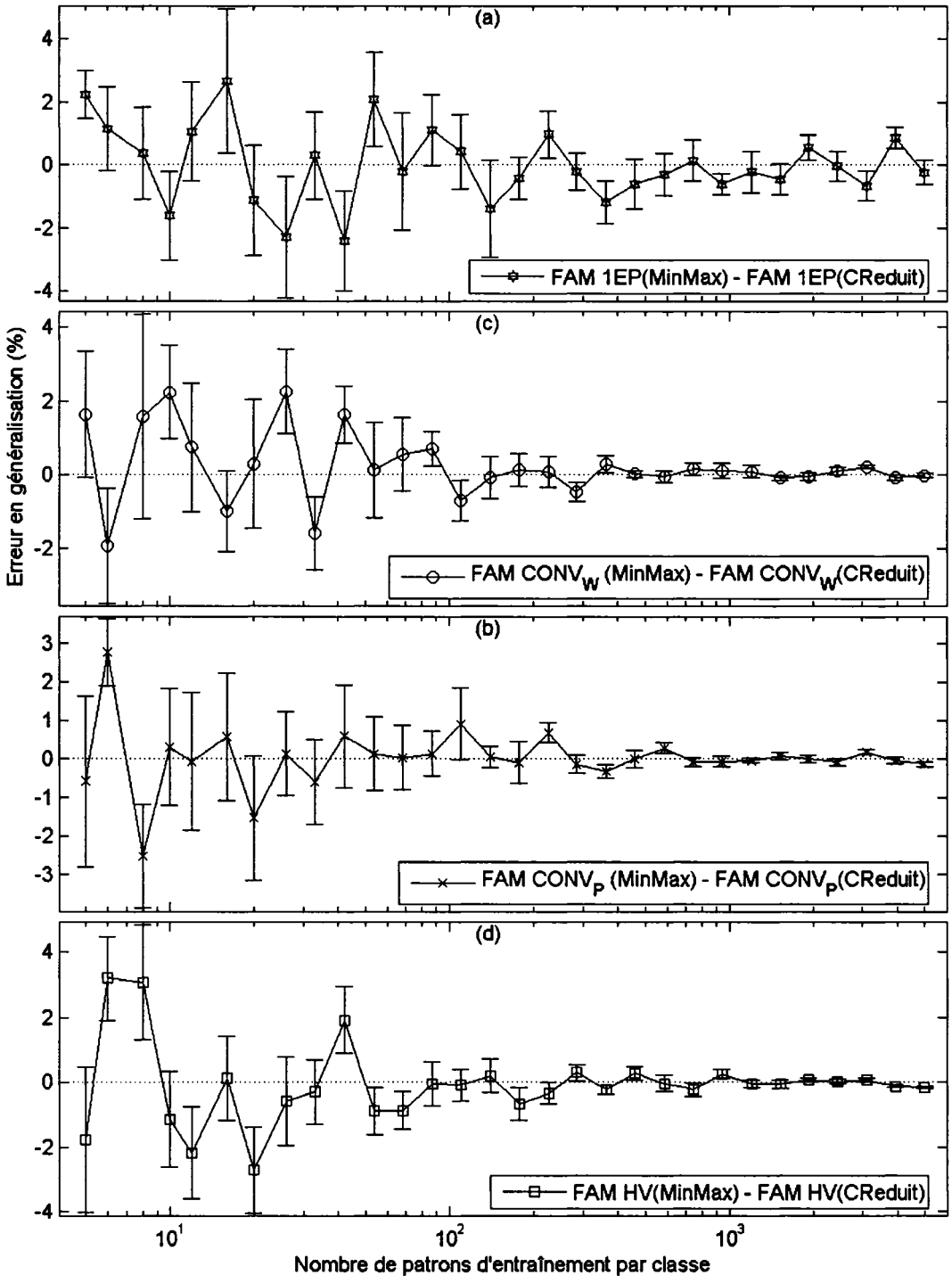


Figure 28 Différence entre l'erreur en généralisation avec la base DB_{CiS}

(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

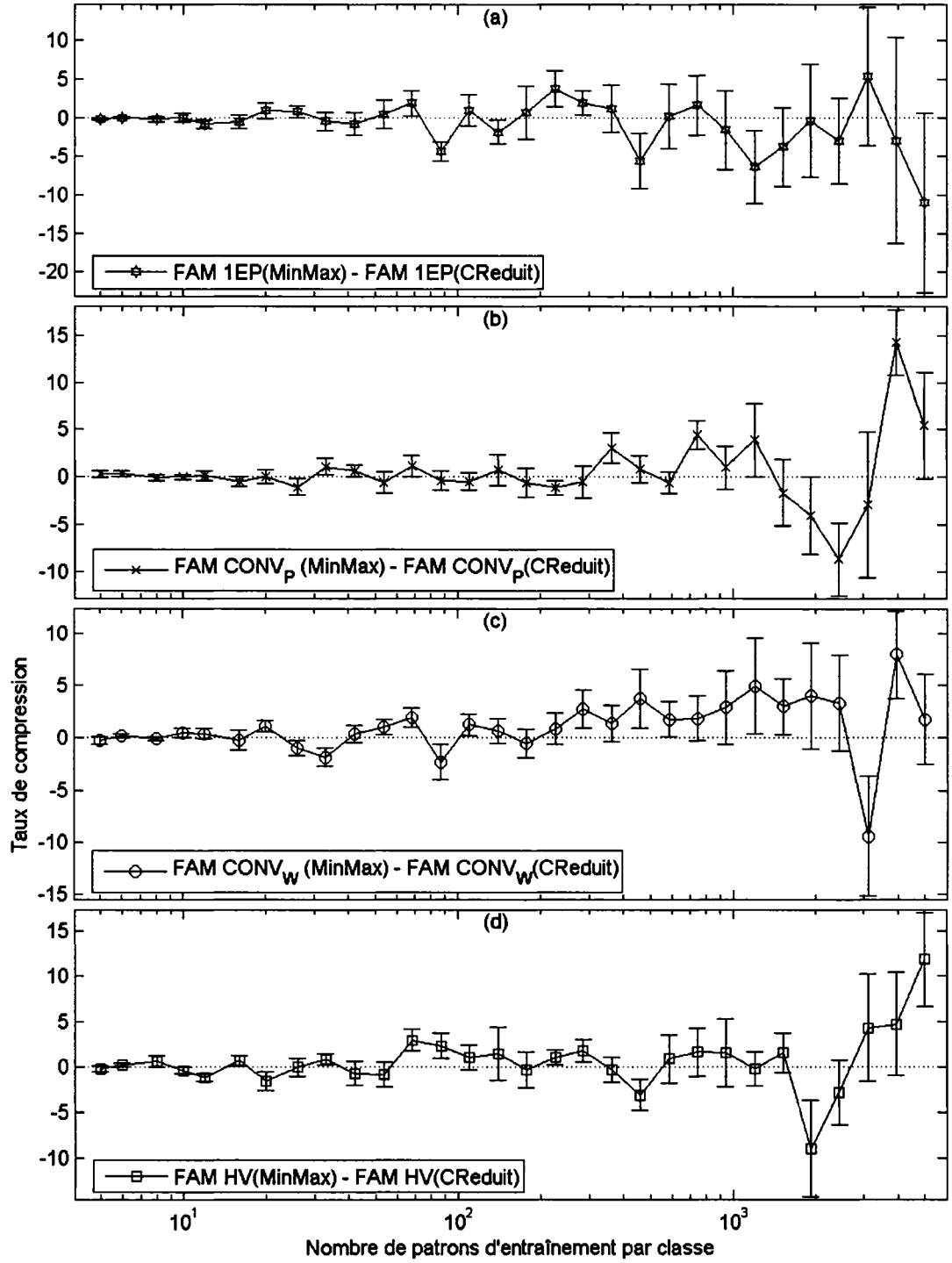


Figure 29 Différence sur le taux de compression avec la base DB_{CIS}

(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

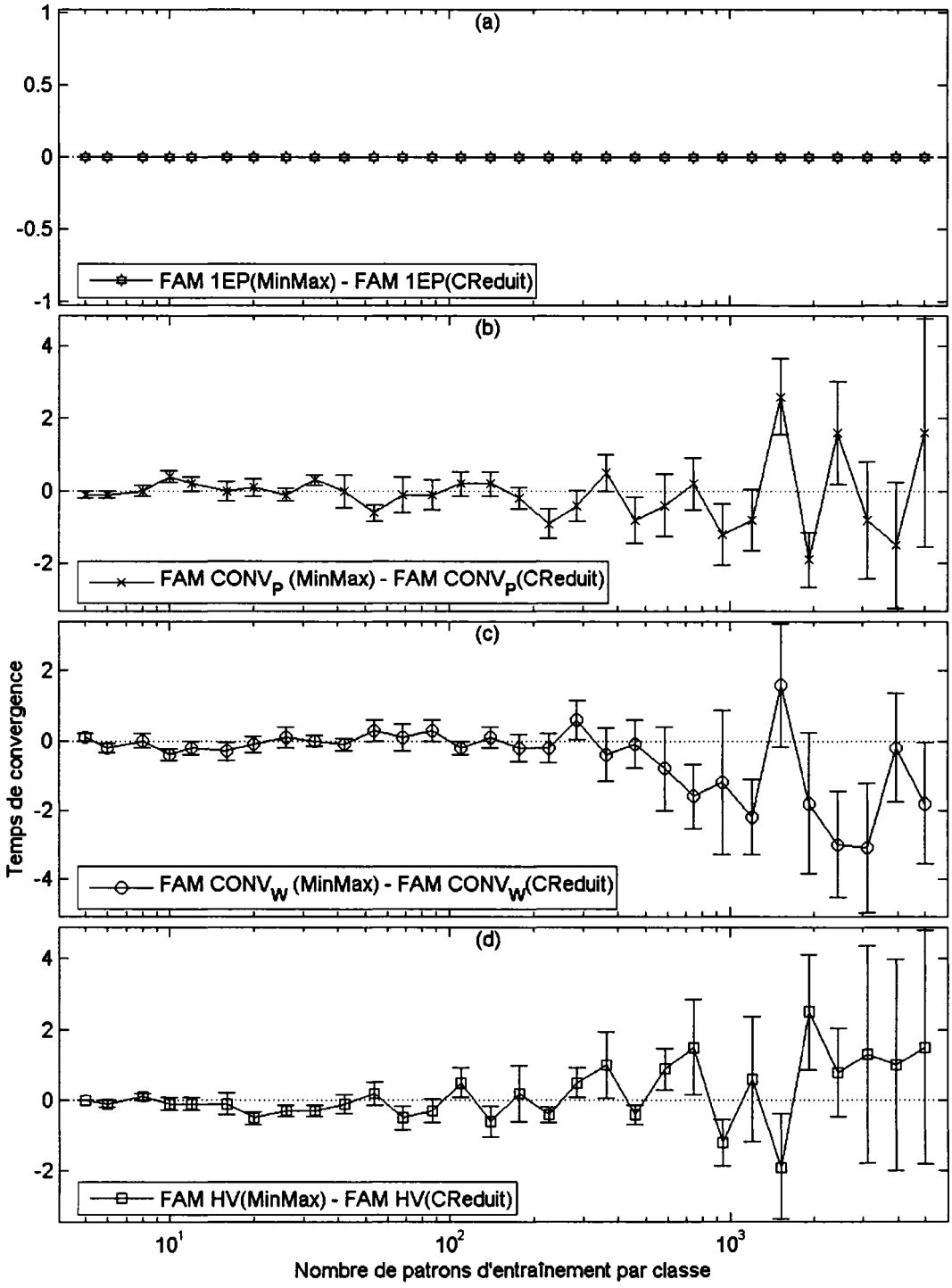


Figure 30 Différence sur le temps de convergence avec la base DB_{CiS}

(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

3.3.3 Analyse

En observant les résultats obtenus, tant avec les bases de données avec chevauchement qu'avec celles sans chevauchement, on constate qu'il n'y a pas de différences significatives entre les deux méthodes de normalisation utilisées.

En effet, les différences entre les deux techniques de normalisation, tant au niveau de l'erreur en généralisation, du taux de compression, que du temps de convergence, sont minimes et ce, pour l'ensemble des degrés de chevauchement.

Aucune des deux techniques de normalisation n'influence de façon significative les performances en généralisation, le temps de convergence et le niveau de compression des réseaux fuzzy ARTMAP sur les bases de données synthétiques que nous avons testées. A partir d'ici, nous n'utiliserons que la technique de normalisation MinMax.

3.4 Effets de la polarité du MatchTracking

Lors de l'utilisation des réseaux fuzzy ARTMAP, les paramètres internes des réseaux doivent être initialisés. Dans toutes les expériences précédentes, nous avons utilisé les paramètres généraux, admis par la communauté scientifique, soit : $\alpha = 0.01$, $\bar{\rho}=0.0$, $\beta = 1.0$ et $\varepsilon = 0.001$. Ces valeurs tendent à maximiser les performances et la compression des réseaux fuzzy ARTMAP. Certaines expériences ont démontré qu'en inversant le signe du paramètre epsilon ($\varepsilon = -0.001$), on peut obtenir de meilleurs résultats. L'inversion de ce paramètre est connue sous le nom de MatchTracking négatif (MT-).

Cette section traite de l'influence de la polarité du MatchTracking, soit la différence entre le MatchTracking positif (MT+) et le MatchTracking négatif (MT-). En comparant les résultats obtenus pour les quatre stratégies d'apprentissage, nous sommes en mesure

de déterminer l'influence de la polarité d'epsilon et son impact sur les performances du réseau fuzzy ARTMAP au niveau de l'erreur en généralisation, du temps de convergence ainsi que du taux de compression. Les résultats obtenus dans cette sous-section ont contribués à la publication d'un article [43].

3.4.1 Bases de données avec chevauchement

Cette sous-section présente la comparaison des résultats entre MT- et MT+ pour la base de données $DB_{\mu}(9\%)$ pour l'erreur en généralisation, le taux de compression et le temps de convergence. Ainsi, lorsque la courbe est positive, la valeur obtenue avec MT- est plus grande que celle obtenue avec la base MT+, et vice-versa. Tel que nous pouvons le voir sur les figures suivantes, le fait de changer la polarité du MT du réseau fuzzy ARTMAP influence la performance en généralisation, le temps de convergence ainsi que le taux de compression des réseaux créés. L'annexe 6 présente ces résultats pour tous les autres degrés de chevauchement.

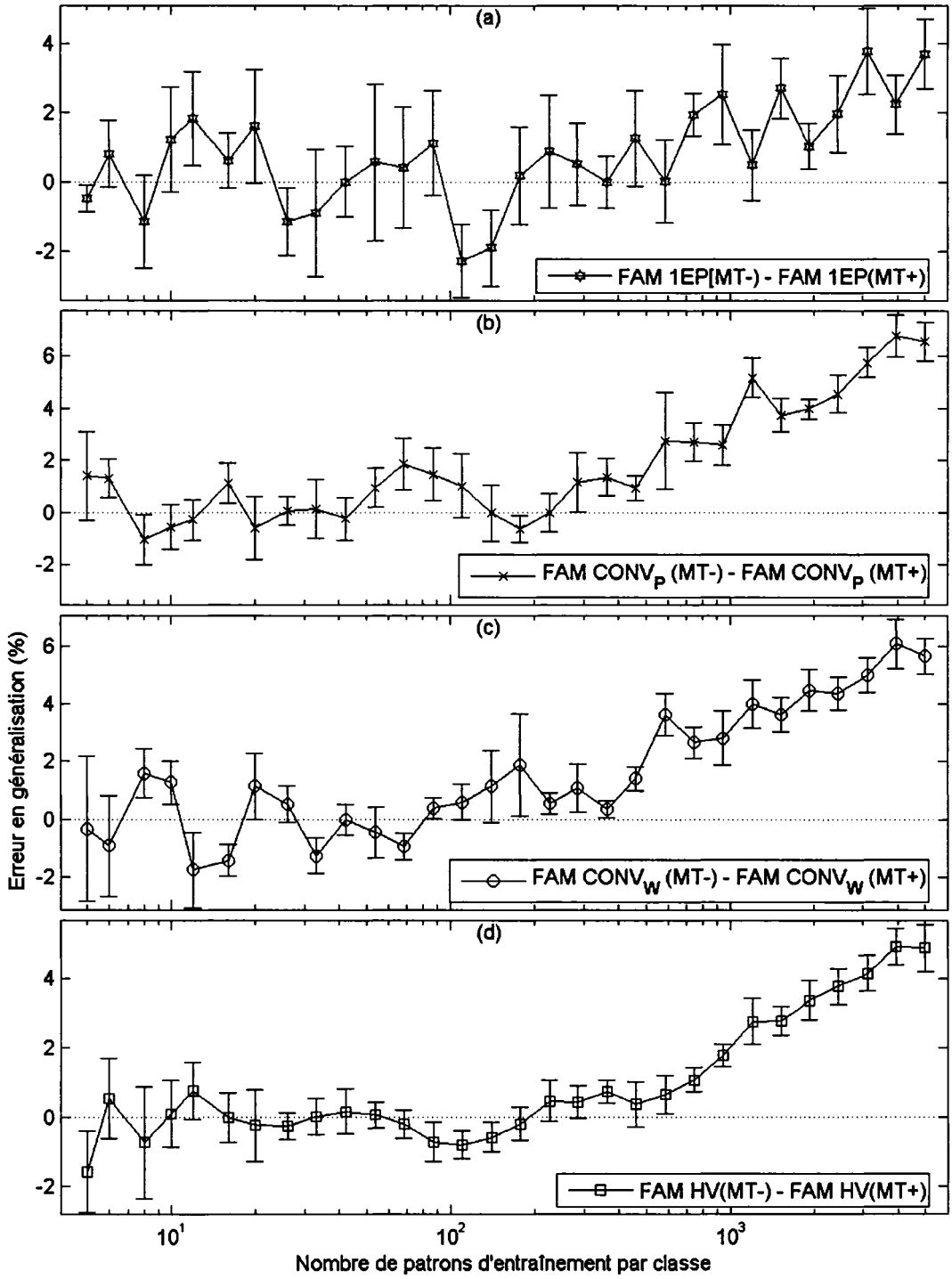


Figure 31 Effet entre MT- et MT+ sur l'erreur en généralisation avec $DB_\mu(9\%)$
(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

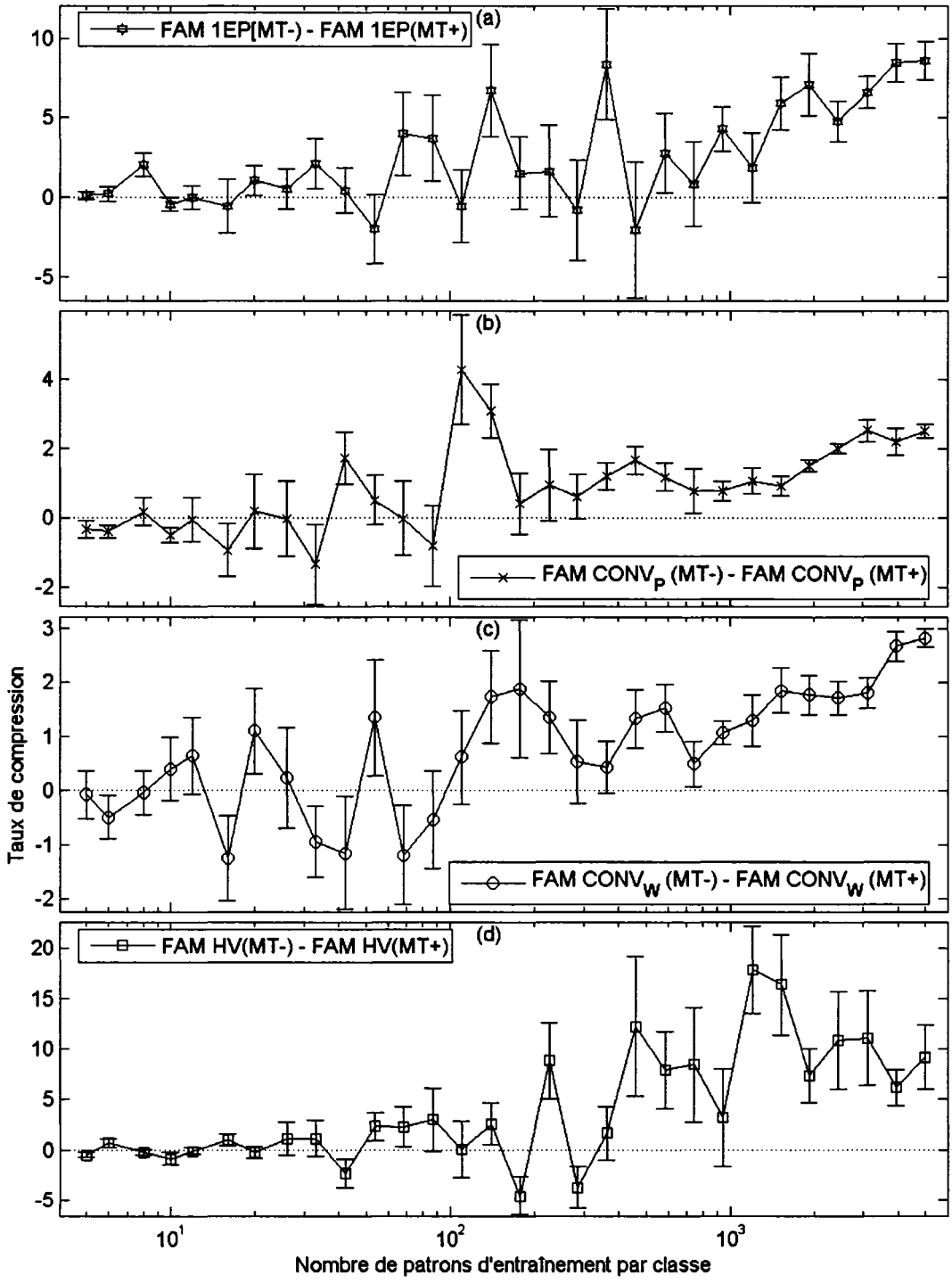


Figure 32 Effet entre MT- et MT+ sur le taux de compression avec $DB_\mu(9\%)$

(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

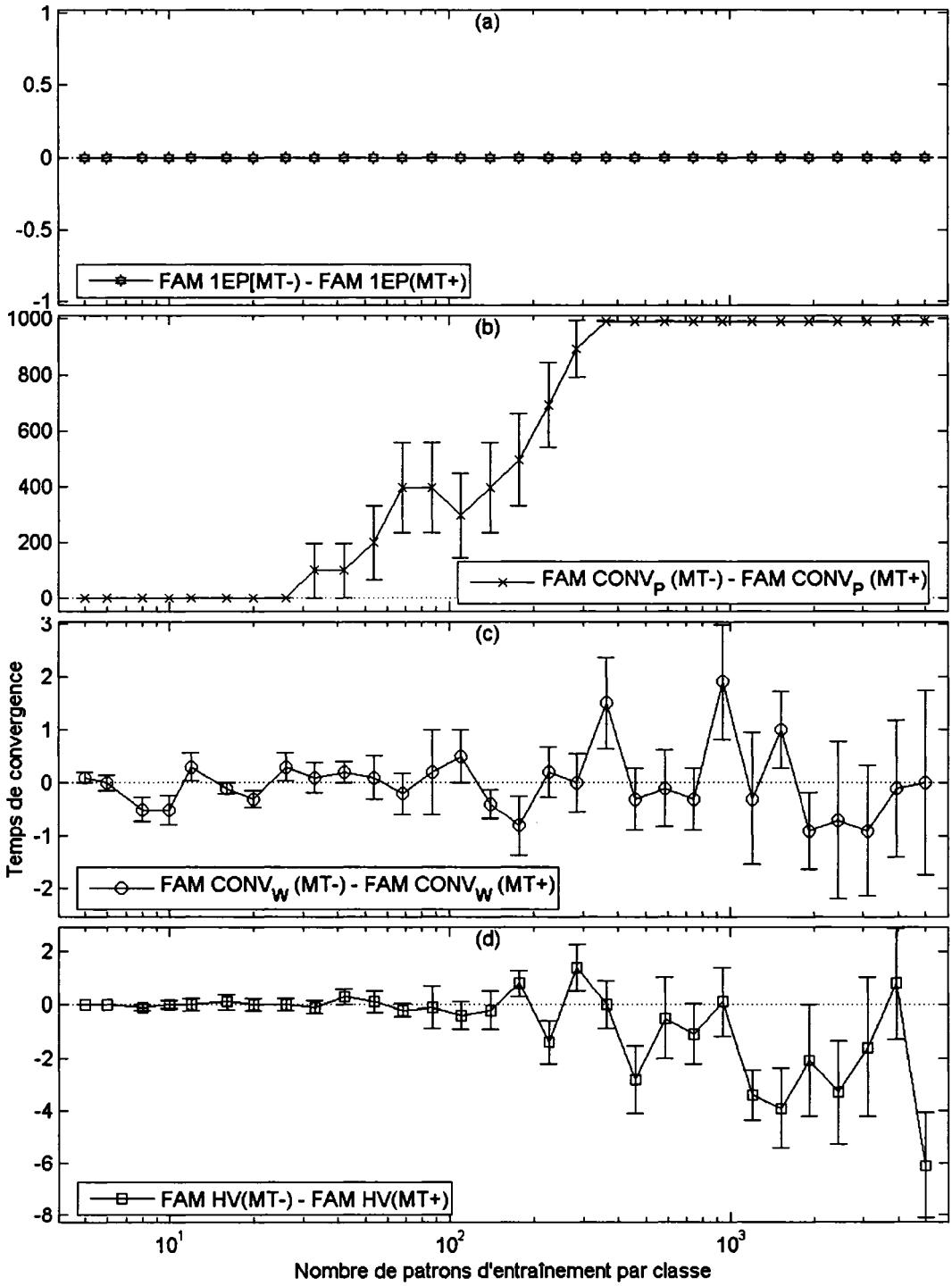


Figure 33 Effet entre MT- et MT+ sur le temps de convergence avec $DB_{\mu}(9\%)$

(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

3.4.2 Bases de données sans chevauchement

Cette sous-section présente la comparaison des résultats entre MT- et MT+ pour la base de donnée DB_{CiS} pour l'erreur en généralisation, le taux de compression et le temps de convergence. La technique de normalisation MinMax est utilisée. Les résultats obtenus avec la base DB_{P2} sont présentés à la fin de l'annexe 6.

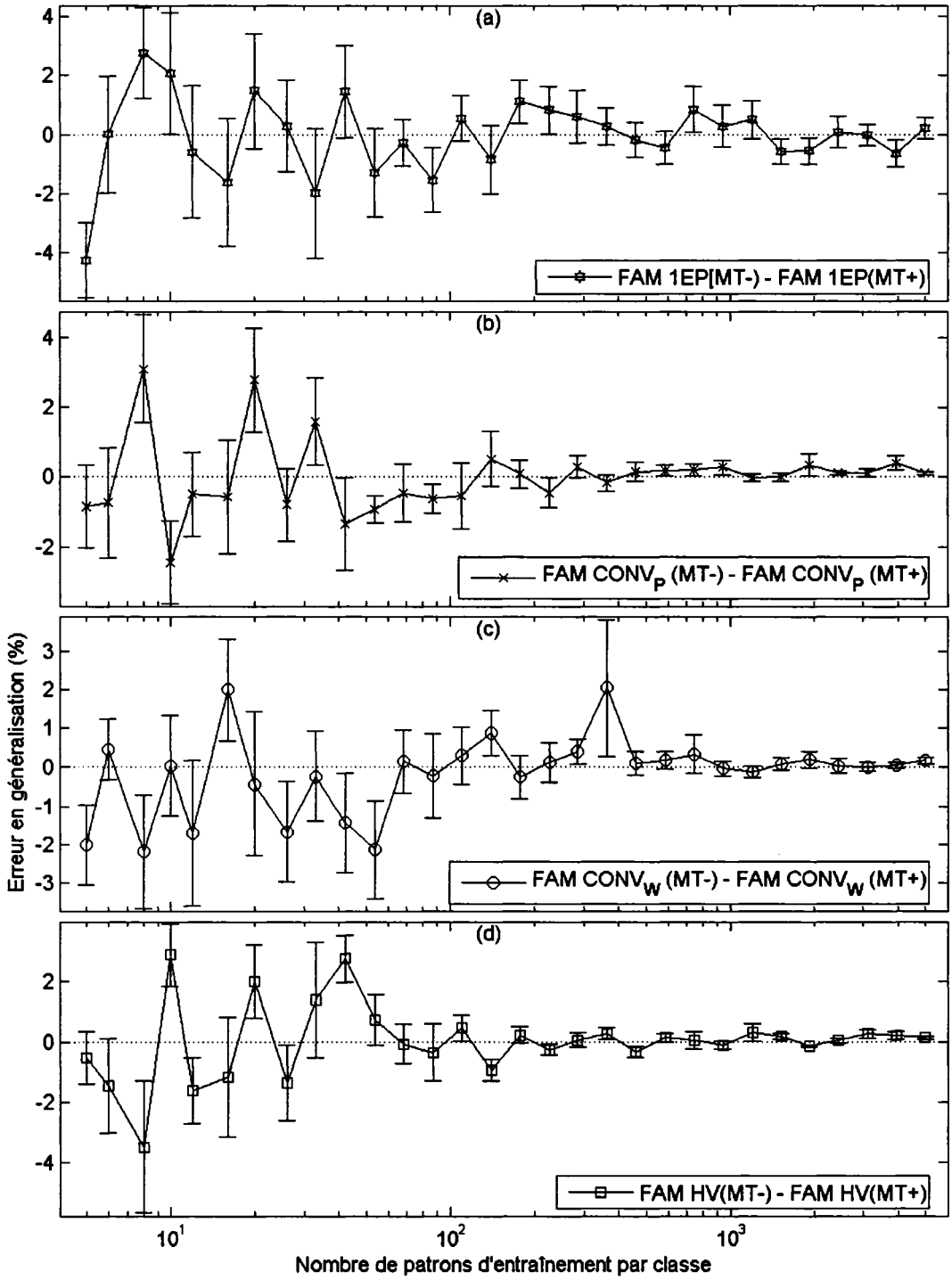


Figure 34 Différence entre MT- et MT+ sur l'erreur en généralisation avec DB_{CIS}
(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

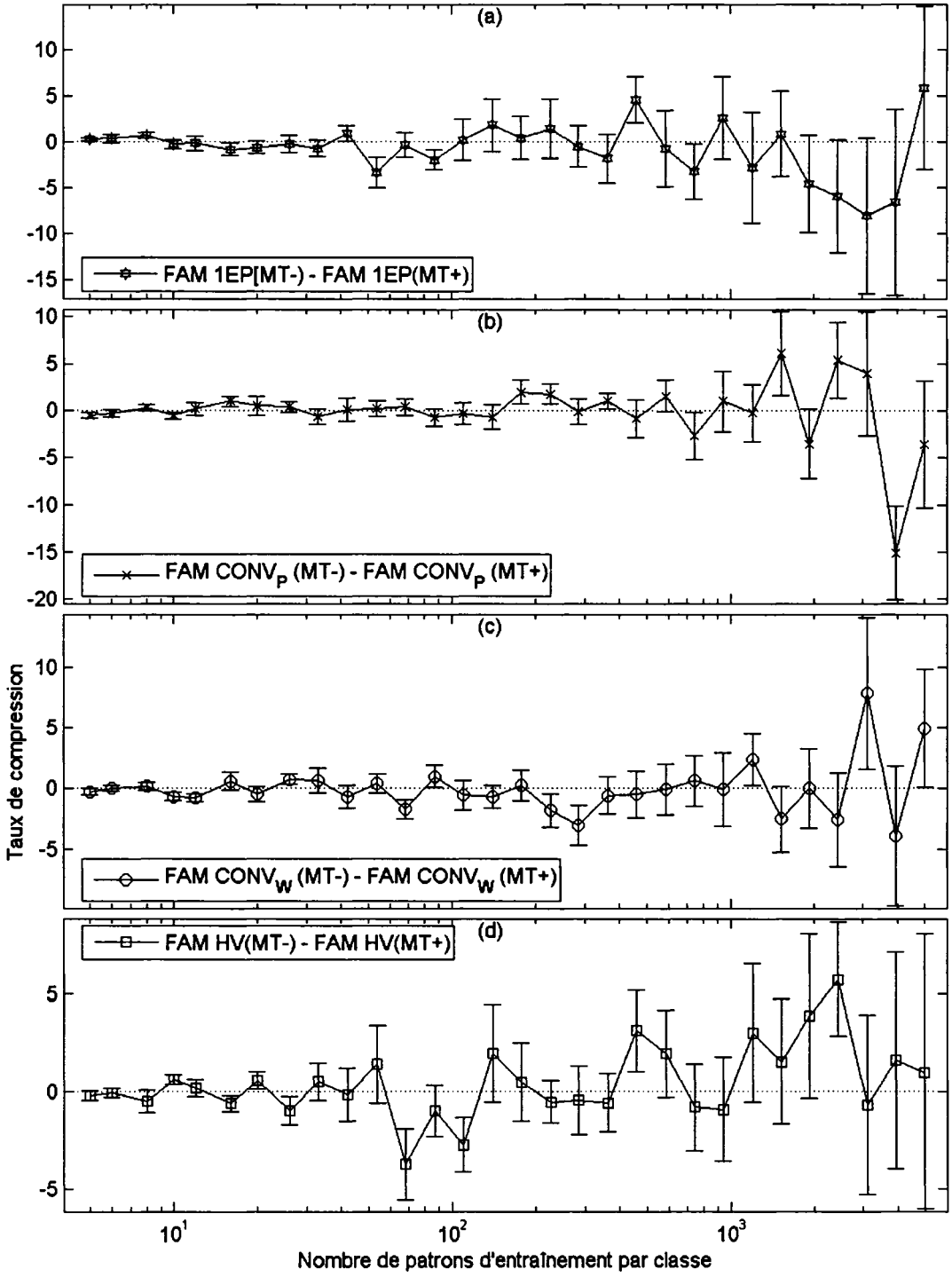


Figure 35 Différence entre MT- et MT+ sur le taux de compression avec DB_{CIS} (a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

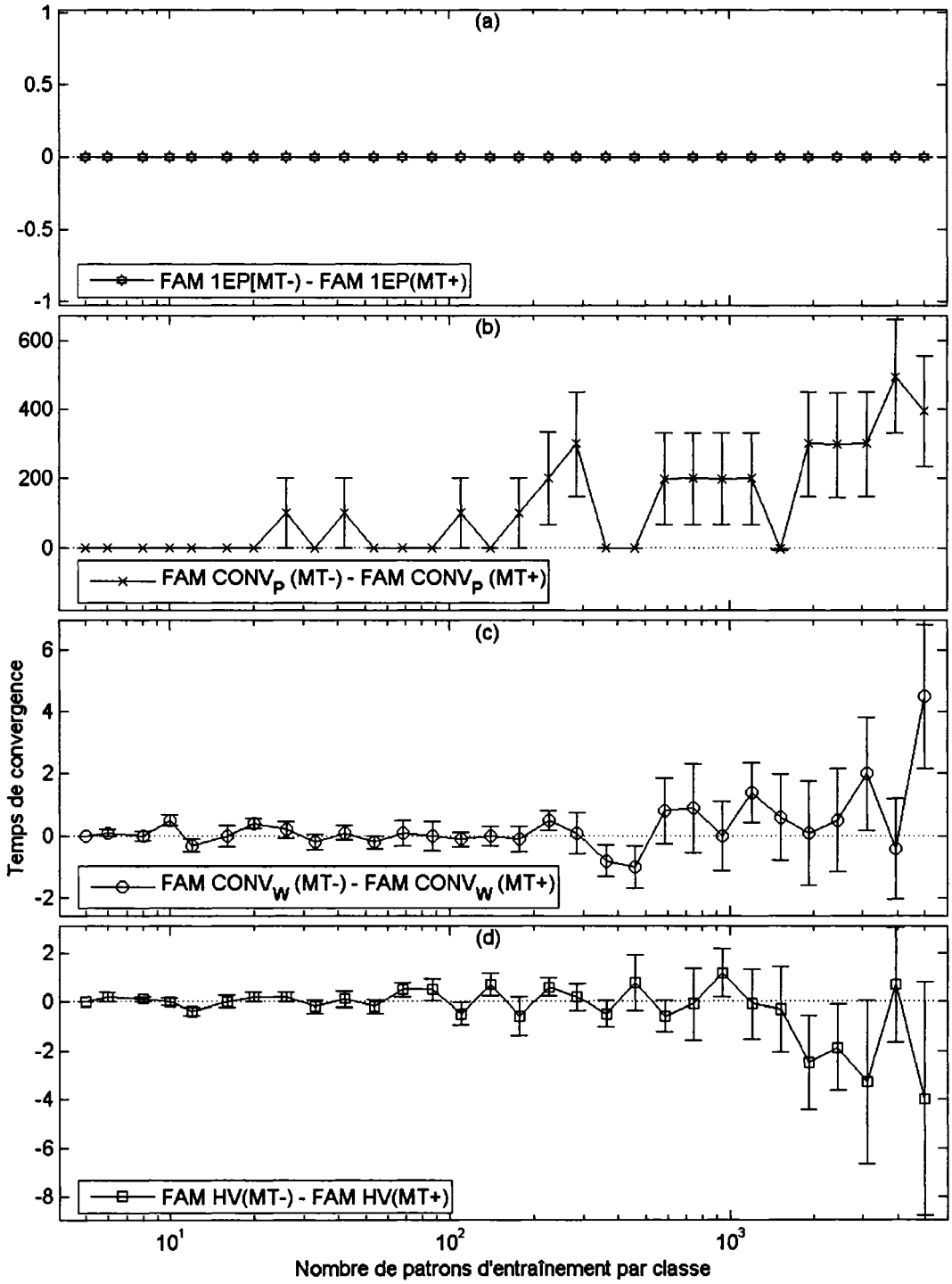


Figure 36 Différence entre MT- et MT+ sur le temps de convergence avec DB_{CIS} (a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

3.4.3 Analyse

Lors de l'utilisation de MT- avec les bases de données avec chevauchement, on remarque que l'erreur en généralisation est plus grande qu'avec MT+ lorsque la taille de la base d'apprentissage augmente. Ainsi, au premier regard, MT+ semble plus approprié pour les bases de données avec chevauchement, car même si MT- obtient de meilleurs taux de compression que MT+, cet avantage est au détriment de la performance en généralisation.

Puisque MT- obtient une plus grande erreur en généralisation avec la taille maximale de la base d'apprentissage, l'erreur de sur-apprentissage engendrée avec ce type de MatchTracking sera plus grande. La figure 37 présente les erreurs de sur-apprentissage pour les quatre méthodes d'entraînement avec MT- ainsi que, comme référence, celles obtenues par la HV avec MT+.

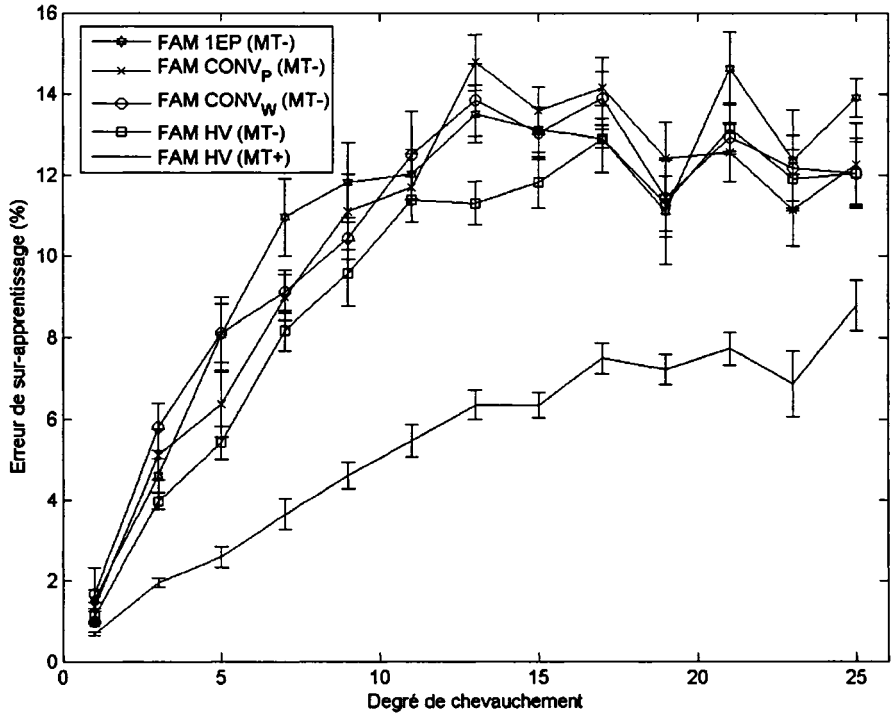


Figure 37 Erreur de sur-apprentissage avec MT-

L'erreur de sur-apprentissage générée par la taille de la base d'apprentissage est bel et bien plus grande avec MT- qu'avec MT+. Mais qu'en est-il de l'erreur nette lorsque la taille de la base d'apprentissage est optimisée? La Figure 38 présente l'erreur nette obtenue avec les bases DB_μ pour la stratégie d'apprentissage HV pour les deux types de MatchTracking lorsque la taille de la base d'entraînement est optimisée et lorsque tous les patrons d'entraînement sont utilisés (5k/ω).

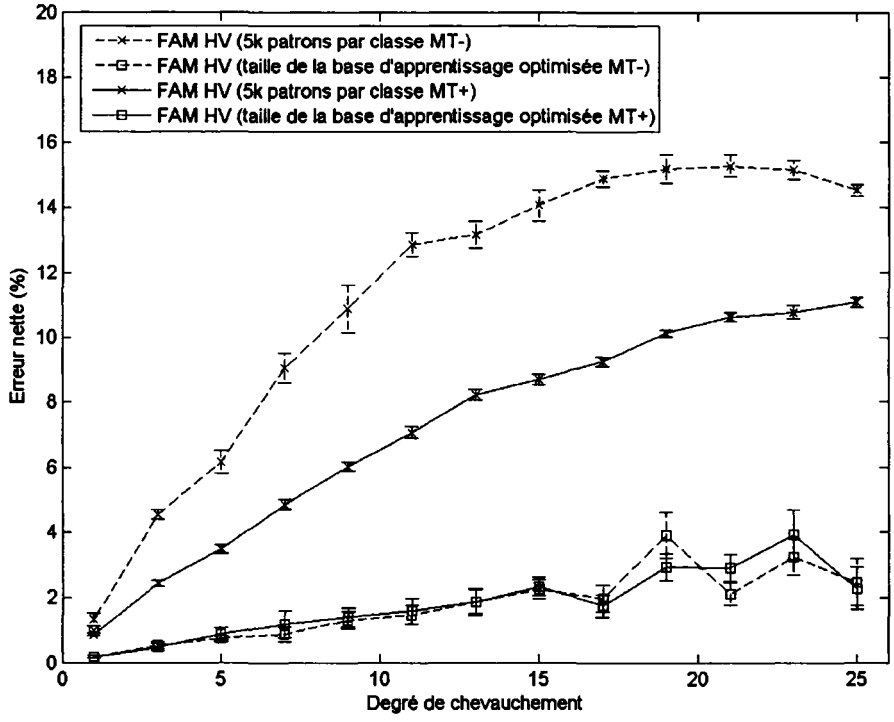


Figure 38 Erreur nette sur DB_μ avec la HV pour MT- et MT+

Tel que nous l'avons déjà remarqué avec les figures 30 et 35, l'erreur en généralisation obtenue lors de l'utilisation de la taille maximale de la base d'apprentissage est supérieure lorsque MT- est utilisé. Ceci se reflète à la Figure 38 par l'écart entre les deux courbes lors de l'utilisation de la taille maximale de la base d'entraînement. Par contre, lorsque la taille de la base d'entraînement est optimisée, la différence entre l'erreur nette obtenue avec MT- et MT+ est pratiquement inexistante.

Ainsi, lorsque la taille de la base d'apprentissage est optimisée les deux types de MatchTracking offrent des performances similaires, mais MT- obtient des taux de compression généralement supérieurs comparativement à MT+. Il faut conclure qu'avec l'optimisation de la taille de la base d'entraînement, MT- est plus performant dans le cas des bases de données avec chevauchement.

Si l'on observe les temps de convergence obtenus avec la stratégie d'apprentissage de convergence des patrons (voir figure 33(b)), nous obtenons souvent le nombre maximum d'époques d'entraînement (1000 époques) lors de l'utilisation de MT-. Ceci indique que le réseau ne peut atteindre le premier critère d'arrêt de cette stratégie d'apprentissage, soit une classification parfaite des patrons de la base d'apprentissage. Pourtant lors de l'utilisation de la même base de données avec un MT+, le réseau en était capable. Voyons plus en détails pourquoi ce phénomène survient.

Prenons une situation d'apprentissage avec la stratégie d'apprentissage $CONV_P$ pour 4 patrons de la base d'apprentissage. La Figure 39 présente les catégories créées après une époque d'entraînement. Deux catégories ont été créées.

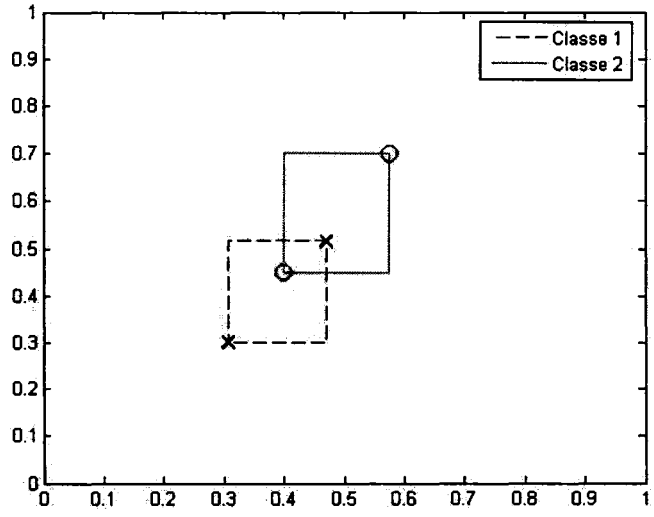


Figure 39 Situation d'apprentissage avec $CONV_P$ pour MT-

La stratégie d'apprentissage par convergence des patrons a, comme condition d'arrêt primaire, la classification parfaite de tous les patrons de la base d'apprentissage. Lorsque le patron de la classe 2 (cercle) qui se trouve dans la catégorie de la classe 1 est testé, le réseau fuzzy ARTMAP cherche, catégorie par catégorie, laquelle représente le mieux ce patron. Puisque ce patron est dans les limites des deux catégories et que la taille de la catégorie appartenant à la classe 1 est légèrement plus petite que l'autre, le réseau sélectionne la catégorie appartenant à la classe 1. Lors de la vérification de la prédiction (phase d'apprentissage), le réseau attribue ce patron à la mauvaise classe. Il élimine donc cette catégorie et regarde si une autre catégorie est capable de classifier ce patron avec la même valeur de résonance obtenue par l'ancienne catégorie plus une petite valeur. Cette petite valeur est en fait epsilon. Puisque l'on utilise MT-, la valeur d'epsilon est négative et la catégorie de la classe 2, qui a une taille légèrement supérieure à celle de la classe 1 est sélectionnée et passera le test de la validation de la prédiction.

Ainsi, le réseau ne change pas la valeur des poids synaptiques et ne crée pas de nouvelles catégories. Mais le critère d'arrêt primaire de la stratégie de convergence des patrons ne sera jamais rencontré dans une telle situation et cette stratégie continuera d'essayer d'apprendre jusqu'à l'obtention du nombre maximal d'époques. Ce phénomène ne survient pas lors de l'utilisation de MT+ car, lorsque la catégorie de la classe 2 est rejetée et que le réseau cherche une autre catégorie presque aussi bonne que la première, il n'en trouve aucune et une nouvelle catégorie est créée pour ce point.

De plus, ce phénomène nous permet de juger de la qualité de la stratégie d'apprentissage $CONV_P$. Bien que cette dernière soit utilisée dans la communauté scientifique, elle n'est pas appropriée pour les réseaux fuzzy ARTMAP. Au lieu d'observer la convergence des patrons, nous devrions regarder la convergence totale des poids synaptiques du réseau. Ainsi, le critère d'arrêt premier de cette stratégie devrait subvenir lorsque deux époques successives obtiennent les mêmes valeurs pour tous les poids synaptiques et non lorsque tous les patrons de la base d'entraînement sont parfaitement classifiés. Le réseau évitera

ainsi d'effectuer des époques d'apprentissage qui ne changent rien aux poids synaptiques du réseau puisque ce dernier a déjà totalement convergé et qu'il ne peut classifier correctement tous les patrons de la base d'entraînement à cause de l'utilisation du MT-.

L'avantage d'utiliser un epsilon négatif (MT-) est de réduire le nombre de catégories créées par le réseau fuzzy ARTMAP, pour ainsi réduire l'effet de la prolifération des catégories et obtenir un meilleur taux de compression. De plus, avec certains types de bases de données, les performances en généralisation sont meilleures avec MT- que lors de l'utilisation de MT+.

Lors de l'utilisation de MT- avec les bases de données sans chevauchement, on remarque que, lorsque la taille de la base d'apprentissage augmente, l'erreur en généralisation obtenue est pratiquement identique à celle obtenue avec MT+. Par contre, lorsqu'on regarde les taux de compression obtenus avec MT- comparativement avec MT+, on remarque qu'il y a beaucoup de variance entre les résultats. Ceci ne nous permet pas de déterminer directement la meilleure polarité du MatchTracking pour les bases de données sans chevauchement. Par contre, nous savons que, généralement, plus le nombre de catégories est élevé lors de l'utilisation des bases sans chevauchement, meilleures sont les performances en généralisation.

Puisqu'il n'y a pratiquement pas de différences entre les erreurs en généralisation obtenues avec MT+ et MT-, on peut en conclure qu'avec les deux types de MatchTracking, aucune dégradation des performances du réseau FAM ne survient avec les bases de données sans chevauchement. Ainsi, plus le nombre de patrons d'entraînement est grand, meilleures sont les performances en généralisation des réseaux fuzzy ARTMAP avec ce type de bases de données. De plus, en examinant le temps de convergence lors de l'utilisation de $CONV_P$, nous remarquons que, lors de l'utilisation de cette stratégie avec MT-, les temps de convergence correspondent également au nombre maximum d'époques d'apprentissage (1000 époques). Ceci provient du même

phénomène que nous venons d'expliquer pour les bases de données avec chevauchement et qui est présenté à la Figure 39.

3.5 Conclusion

Selon les résultats présentés dans ce chapitre, il est clair que les réseaux fuzzy ARTMAP peuvent subir une dégradation des performances lors de l'utilisation des bases de données ayant du chevauchement. Cette dégradation peut être engendrée par le nombre d'époques d'entraînement ainsi que le nombre de patrons d'entraînement. Lors de la comparaison (voir figure 38) entre l'erreur nette avec le nombre d'époques optimisé (HV) et l'erreur nette avec l'optimisation de ces deux facteurs (HV + nombre de patrons d'entraînement), on peut en déduire qu'entre ces deux facteurs, le nombre de patrons d'entraînement est celui qui dégrade le plus les performances des réseaux fuzzy ARTMAP.

Nous avons également remarqué que MT- est plus approprié pour les bases de données avec degré de chevauchement que MT+ lorsque la taille de la base d'apprentissage est optimisée. Cependant, pour les bases de données sans chevauchement, nous ne sommes pas en mesure de favoriser l'un ou l'autre des deux types de MatchTracking.

De plus, il a été démontré que les deux techniques de normalisation testées obtiennent des résultats équivalents, tout comme les deux structures utilisées dans les bases possédant des degrés de chevauchement.

Finalement, tel que nous l'avions envisagé lors de l'utilisation des bases de données sans chevauchement, la structure de ces bases de données influence les performances des réseaux fuzzy ARTMAP. La base DB_{P2} obtient des résultats plus faibles que la base DB_{CiS} car ses frontières de décision sont plus longues et plus complexes que celles de DB_{CiS}.

Il est à noter que la variance des résultats est parfois grande. Cette variance est majoritairement due au nombre de patrons d'apprentissage. Cependant, l'ordre de présentation des patrons ainsi que la valeur des patrons contenue dans les dix différentes bases générées pour chaque cas influencent également la variance des résultats.

Le tableau IX présente un sommaire des résultats obtenus lors de l'utilisation de la taille maximale de la base d'apprentissage avec les quatre stratégies d'entraînement pour l'ensemble des bases de données. Ces résultats sont l'erreur en généralisation moyenne et la dispersion des résultats sur les 10 réplications. L'annexe 7 présente une synthèse des résultats pour toutes les bases de données.

Tableau IX

Résultats sommaires avec 5k patrons par classe

Stratégies d'apprentissage	Erreur en généralisation moyenne (dispersion des résultats) %				
	DB_μ(1%)	DB_μ (9%)	DB_μ (25%)	DB_{CIS}	DB_{P2}
Erreur théorique	1,00	9,00	25,00	0,00	0,00
CQB	1,00 (0,04)	9,12 (0,08)	25,11 (0,10)	ND	ND
***k*NN**	1,08 (0,03)	9,88 (0,08)	27,23 (0,12)	0,86 (0,03)	1,65 (0,04)
1NN	1,54 (0,03)	13,35 (0,10)	33,49 (0,16)	0,84 (0,02)	1,61 (0,04)
FAM 1EP MT-	2,75 (0,20)	22,49 (0,87)	40,58 (0,47)	4,20 (0,25)	8,89 (0,44)
FAM HV MT-	2,17 (0,08)	20,80 (0,56)	39,83 (0,31)	1,69 (0,07)	4,26 (0,16)
FAM $CONV_W$ MT-	2,74 (0,18)	20,45 (0,46)	40,31 (0,27)	1,77 (0,09)	4,48 (0,38)
FAM $CONV_P$ MT-	2,56 (0,09)	22,00 (0,66)	40,42 (0,31)	1,59 (0,04)	4,51 (0,19)
FAM 1EP MT+	2,51 (0,14)	18,78 (0,38)	38,81 (0,36)	3,98 (0,21)	7,33 (0,33)
FAM HV MT+	1,88 (0,05)	15,17 (0,13)	36,10 (0,20)	1,58 (0,05)	3,68 (0,07)
FAM $CONV_W$ MT+	1,97 (0,09)	15,30 (0,16)	35,94 (0,15)	1,64 (0,05)	3,66 (0,08)
FAM $CONV_P$ MT+	1,90 (0,07)	15,44 (0,15)	36,14 (0,20)	1,47 (0,05)	3,61 (0,08)

CHAPITRE 4

STRATÉGIES D'APPRENTISSAGE SPÉCIALISÉES BASÉES SUR L'OPTIMISATION DES PARAMÈTRES DU RÉSEAU FUZZY ARTMAP ÉVALUÉES SUR LES BASES SYNTHÉTIQUES

Lors de notre première réflexion sur la performance des réseaux fuzzy ARTMAP, nous avons identifié plusieurs caractéristiques qui, selon nous, pourraient entraîner la dégradation des performances. Ces caractéristiques sont :

a. le nombre d'époques d'entraînement;
b. la taille de la base d'entraînement;
c. la technique de normalisation;
d. la structure (dispersion et chevauchement) des données générées;
e. l'ordre de présentation des patrons d'entraînement;
f. le type de MatchTracking.

L'analyse des résultats présentés au chapitre 3 a démontré que le nombre d'époques d'entraînement et la taille de la base de données d'apprentissage pouvaient engendrer une dégradation des performances due au phénomène de sur-apprentissage pour les bases avec chevauchement (DB_μ et DB_σ). Par contre, la technique de normalisation ainsi que la structure des bases de données n'ont aucune incidence sur la dégradation des performances dans les réseaux fuzzy ARTMAP. L'impact de l'ordre de présentation des patrons d'entraînement n'est pas directement abordé, car ce sujet est très vaste. Par contre, l'optimisation de la taille de la base d'entraînement semble atténuer la dégradation des performances causée par cette caractéristique.

La polarité du MatchTracking utilisée fait varier l'erreur de sur-apprentissage de manière significative pour les bases de données avec chevauchement. En analysant ces résultats, il faut s'interroger sur l'effet combiné des quatre paramètres internes des réseaux fuzzy ARTMAP, soit: le paramètre de choix (α), de vitesse d'apprentissage (β), de MatchTracking (ε) et de vigilance de base ($\overline{\rho}$). En effet, si l'erreur en généralisation

peut varier en changeant la polarité d'epsilon (MT- vs MT+), il doit exister une combinaison de paramètres permettant d'optimiser la performance en généralisation. Dans le même ordre d'idées, l'erreur nette restante après l'optimisation de la taille de la base d'apprentissage (voir figure 38) peut-elle être réduite avec une meilleure sélection des paramètres? La dégradation des performances due à la taille de la base d'apprentissage et au nombre d'époques d'entraînement sont-elles influencées par les valeurs des paramètres utilisées? Pour répondre à ces questions et ainsi analyser l'impact des valeurs des paramètres internes sur la performance en généralisation du FAM, nous utilisons les stratégies d'apprentissage spécialisées pour FAM, avec l'algorithme PSO, que nous avons développées.

Ces stratégies permettent de trouver des valeurs pour chacun des paramètres dans le but d'optimiser les performances en généralisation des réseaux fuzzy ARTMAP, et ce, peu importe la base de données utilisée. Ainsi, ce chapitre analyse les effets des paramètres internes du FAM lorsque ceux-ci sont optimisés avec l'algorithme PSO par nos stratégies d'apprentissage spécialisées, pour les bases de données synthétiques. Les résultats obtenus dans ce chapitre ont contribué à la publication d'un article [35].

À noter que les temps de convergence incluent toutes les époques effectuées par toutes les particules lors de l'optimisation PSO. Le temps de convergence varie entre 660 époques (15 particules PSO, 11 itérations PSO, 4 répétitions PSO, et 1 époques d'entraînement FAM) et $6x10^6$ époques (15 particules PSO, 100 itérations PSO, 4 répétitions PSO, et 1000 époques d'entraînement FAM).

4.1 Bases de données avec chevauchement

Cette section traite des effets de l'optimisation des paramètres internes du réseau fuzzy ARTMAP avec les bases de données DB_μ et DB_σ. Étant donné qu'aucune différence n'a

été remarquée entre les deux techniques de normalisation (voir section 3.3), la normalisation MinMax est utilisée.

4.1.1 Résultats

Les figures suivantes présentent les erreurs en généralisation, les temps de convergence ainsi que les taux de compression obtenus pour les quatre stratégies d'apprentissage et les bases de données $DB_{\mu}(1\%)$, $DB_{\mu}(9\%)$, $DB_{\mu}(25\%)$ et $DB_{\sigma}(9\%)$. La stratégie d'apprentissage HV*(MT+)*, le classificateur quadratique Bayesien (CQB) ainsi que le *k*NN sont également présentés comme référence. De plus, les valeurs optimisées des quatre paramètres internes des réseaux fuzzy ARTMAP sont également analysées.

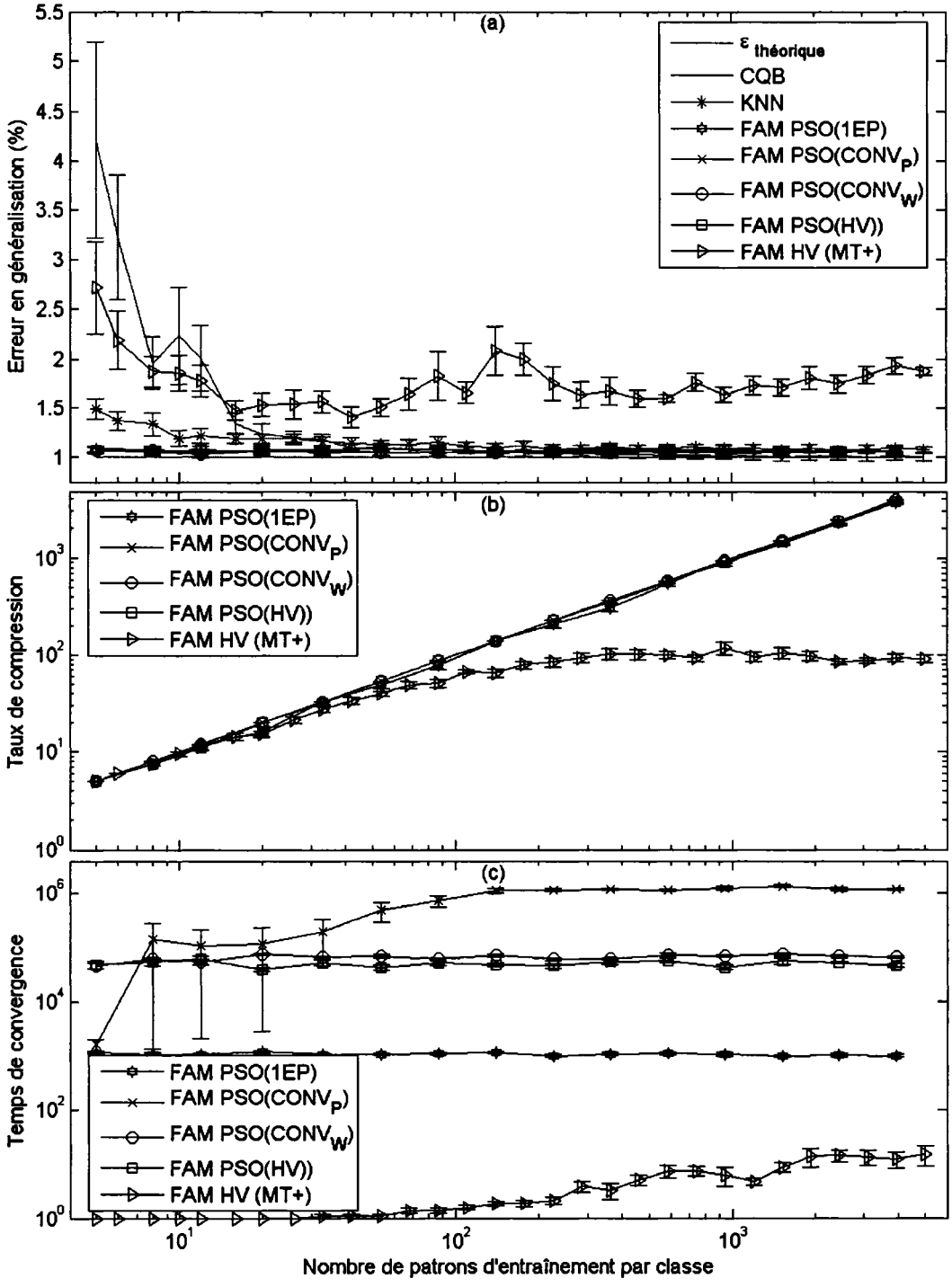


Figure 40 Performance du FAM avec les stratégies PSO sur la base $DB_{\mu}(1\%)$
(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

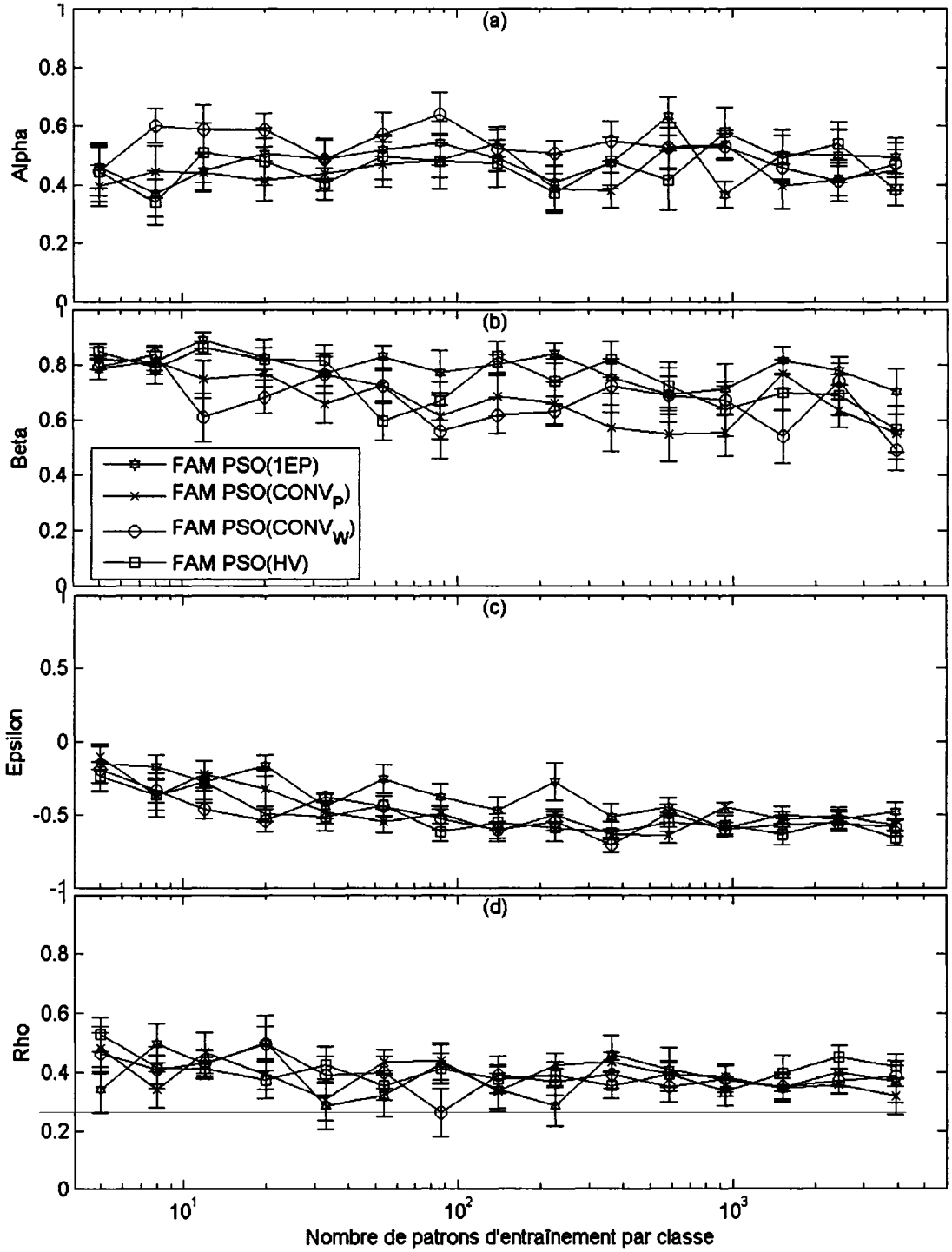


Figure 41 Valeurs des paramètres internes du FAM lors de l'utilisation des stratégies spécialisées PSO avec la base $DB_\mu(1\%)$

(a) α: paramètre de choix, (b) β : vitesse d'apprentissage, (c) ε : paramètre de MatchTracking et (d) $\overline{\rho}$: vigilance de base

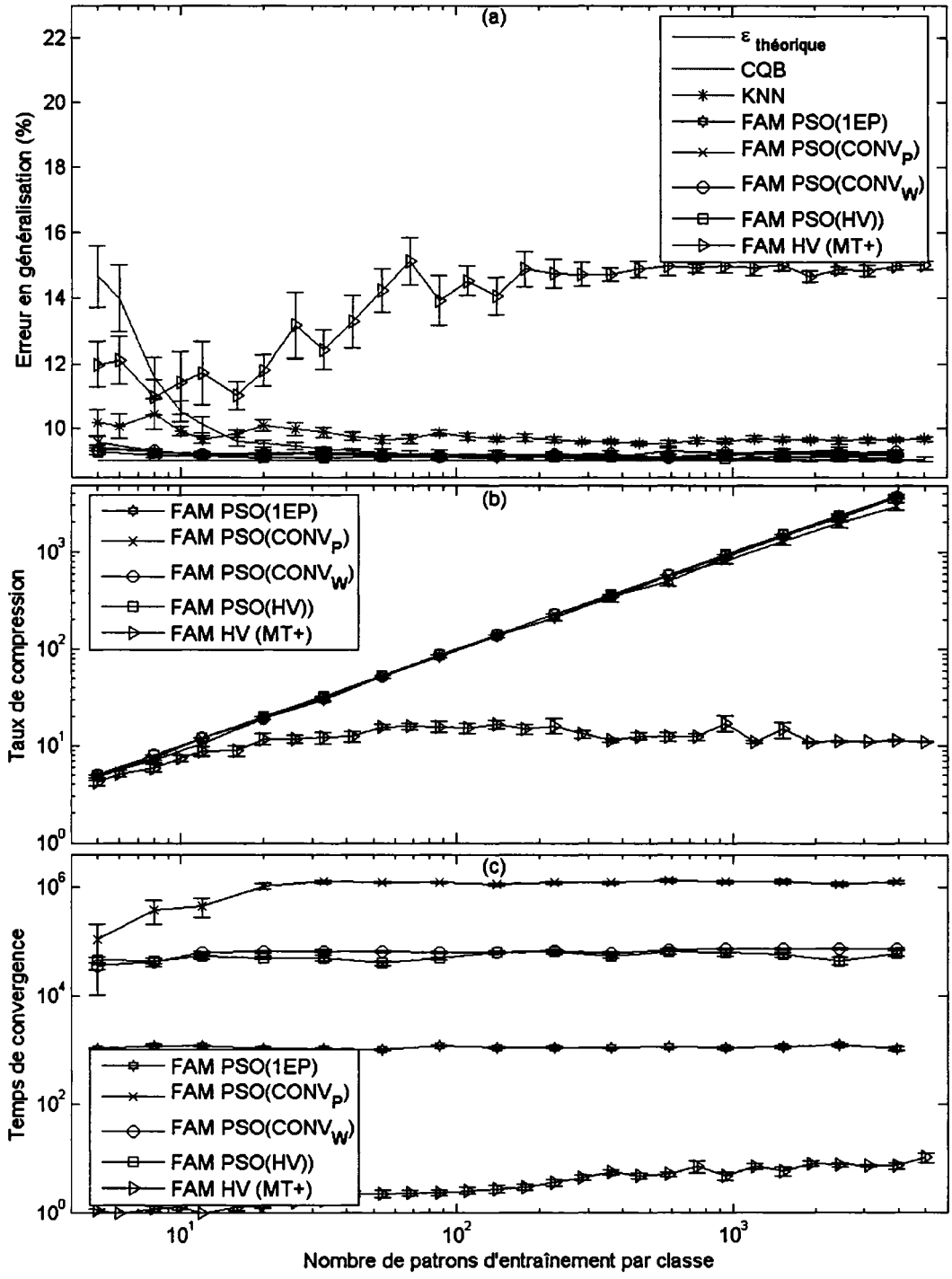


Figure 42 Performance du FAM avec les stratégies PSO sur la base $DB_{\mu}(9\%)$
(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

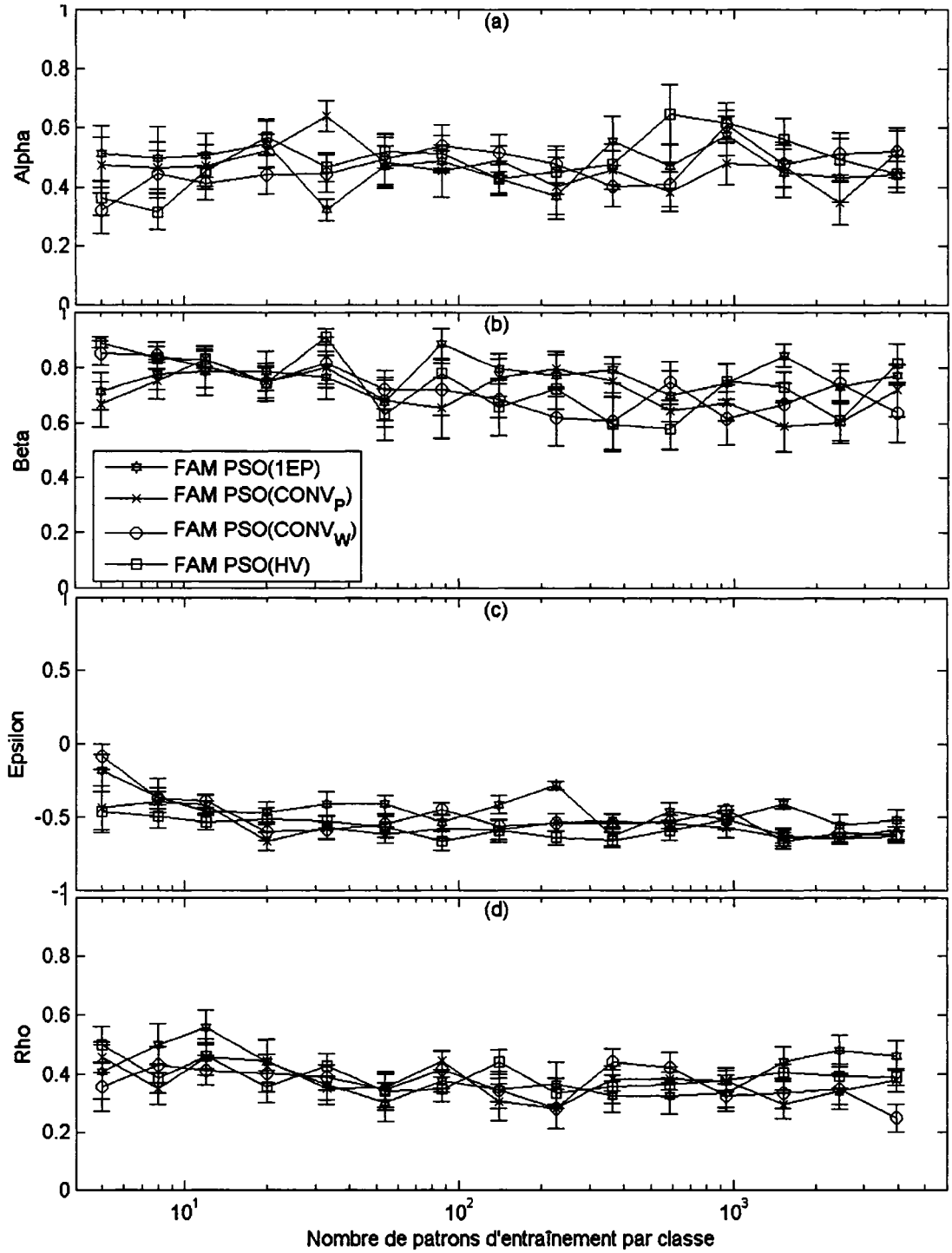


Figure 43 Valeurs des paramètres internes du FAM lors de l'utilisation des stratégies spécialisées PSO avec la base $DB_{\mu}(9\%)$

(a) α: paramètre de choix, (b) β : vitesse d'apprentissage, (c) ε : paramètre de MatchTracking et (d) $\bar{\rho}$: vigilance de base

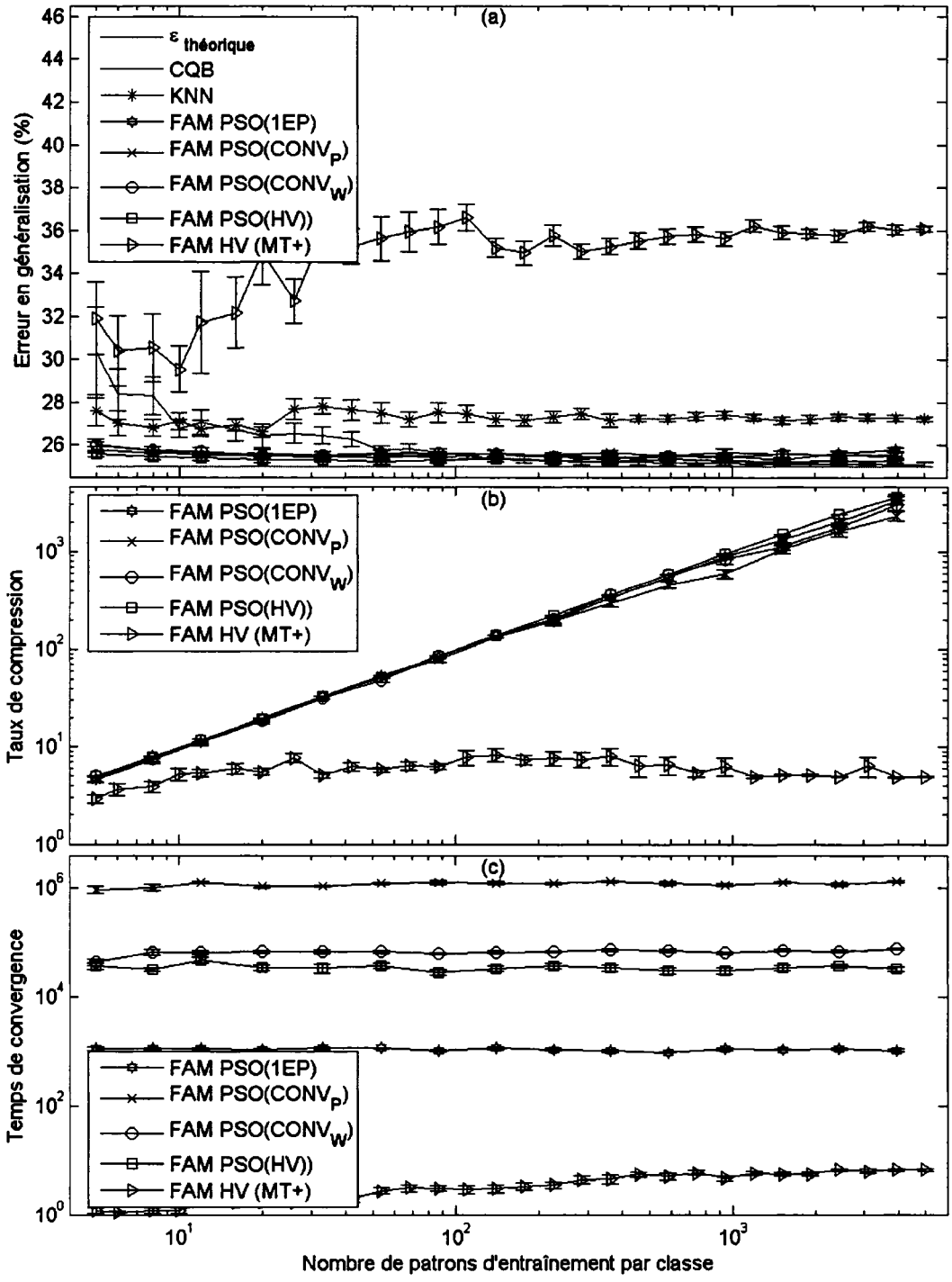


Figure 44 Performance du FAM avec les stratégies PSO sur la base $DB_{\mu}(25\%)$ (a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

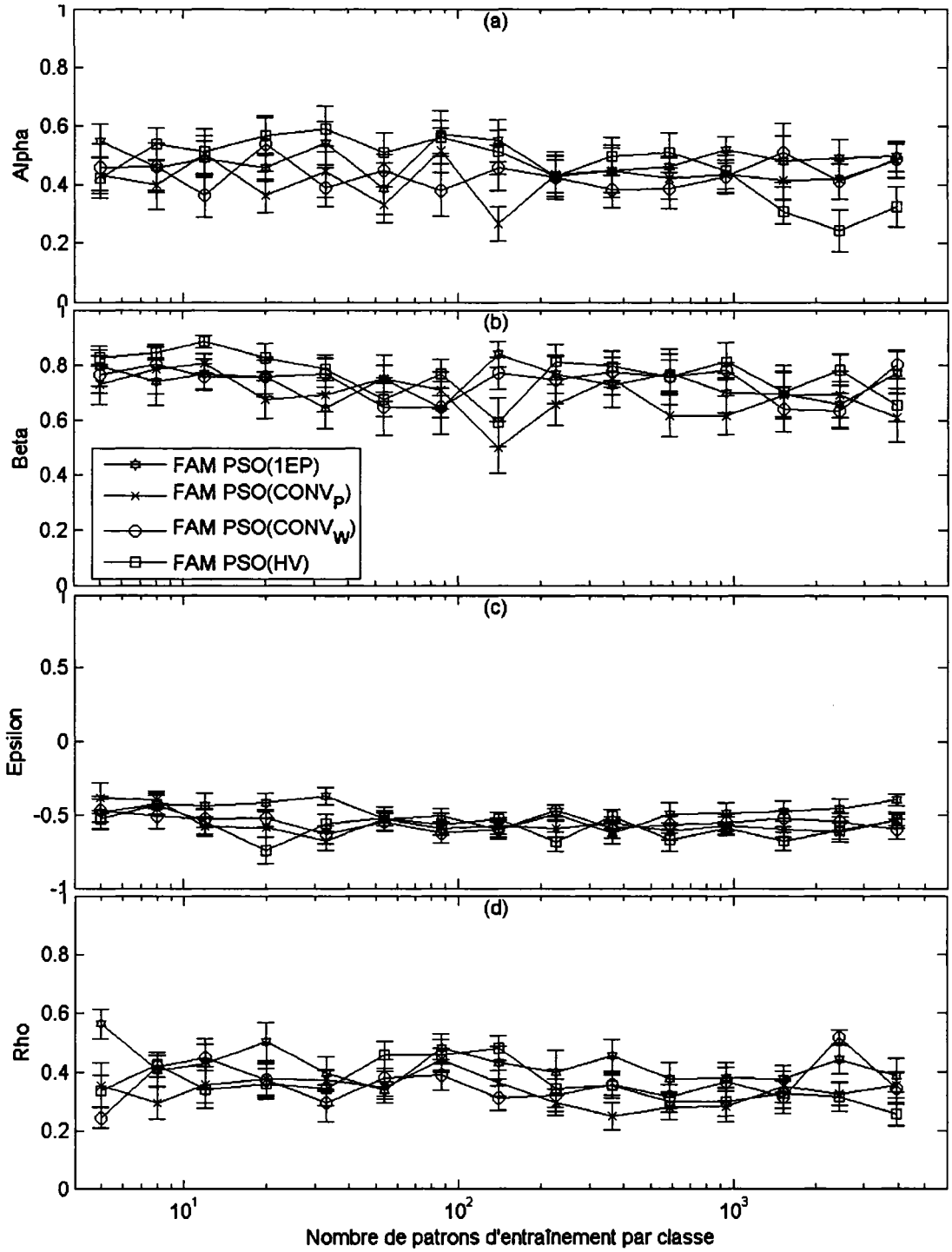


Figure 45 Valeurs des paramètres internes du FAM lors de l'utilisation des stratégies spécialisées PSO avec la base $DB_\mu(25\%)$

(a) α: paramètre de choix, (b) β : vitesse d'apprentissage, (c) ε : paramètre de MatchTracking et (d) $\overline{\rho}$: vigilance de base

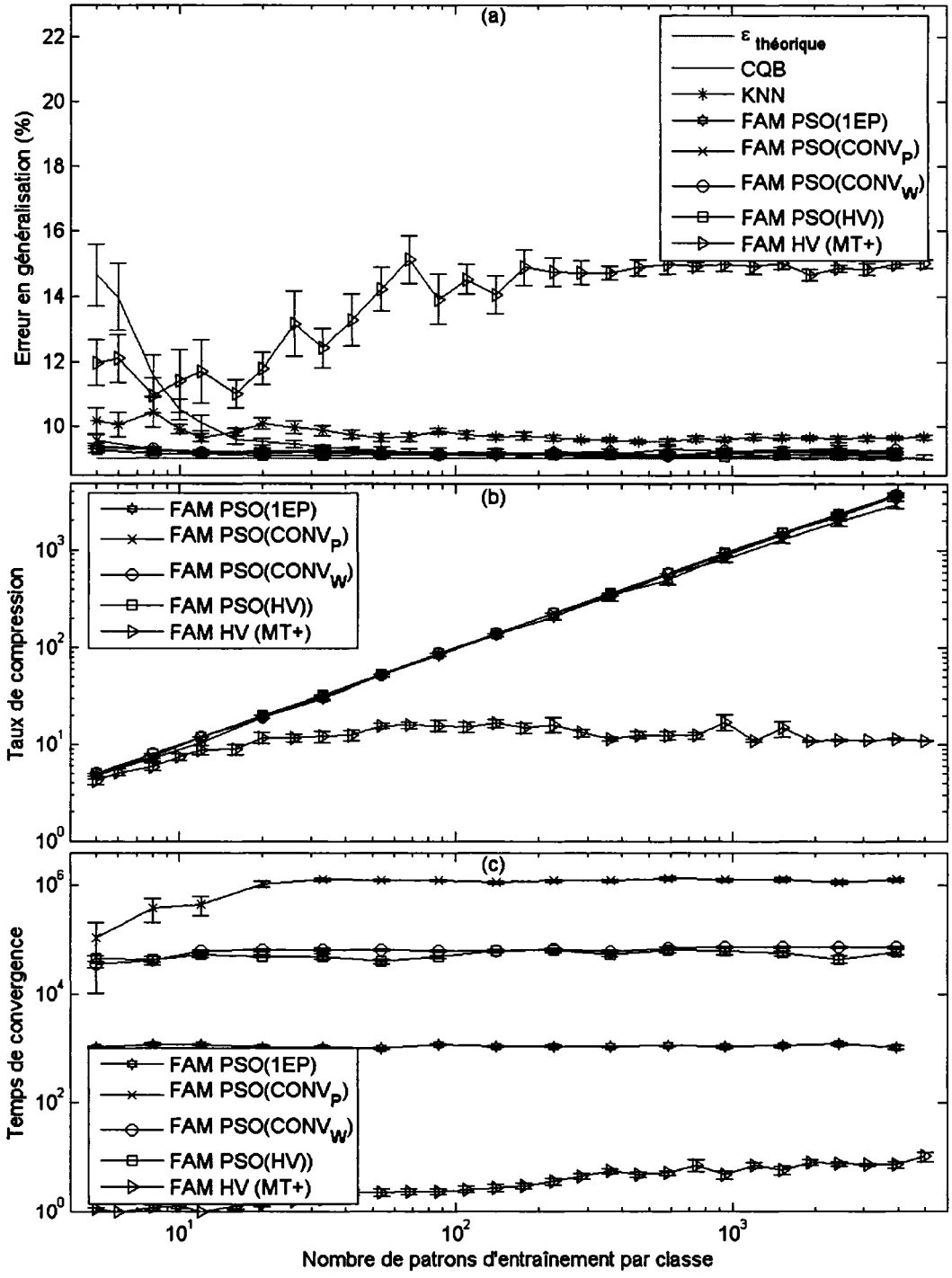


Figure 46 Performance du FAM avec les stratégies PSO sur la base $DB_\sigma(9\%)$

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

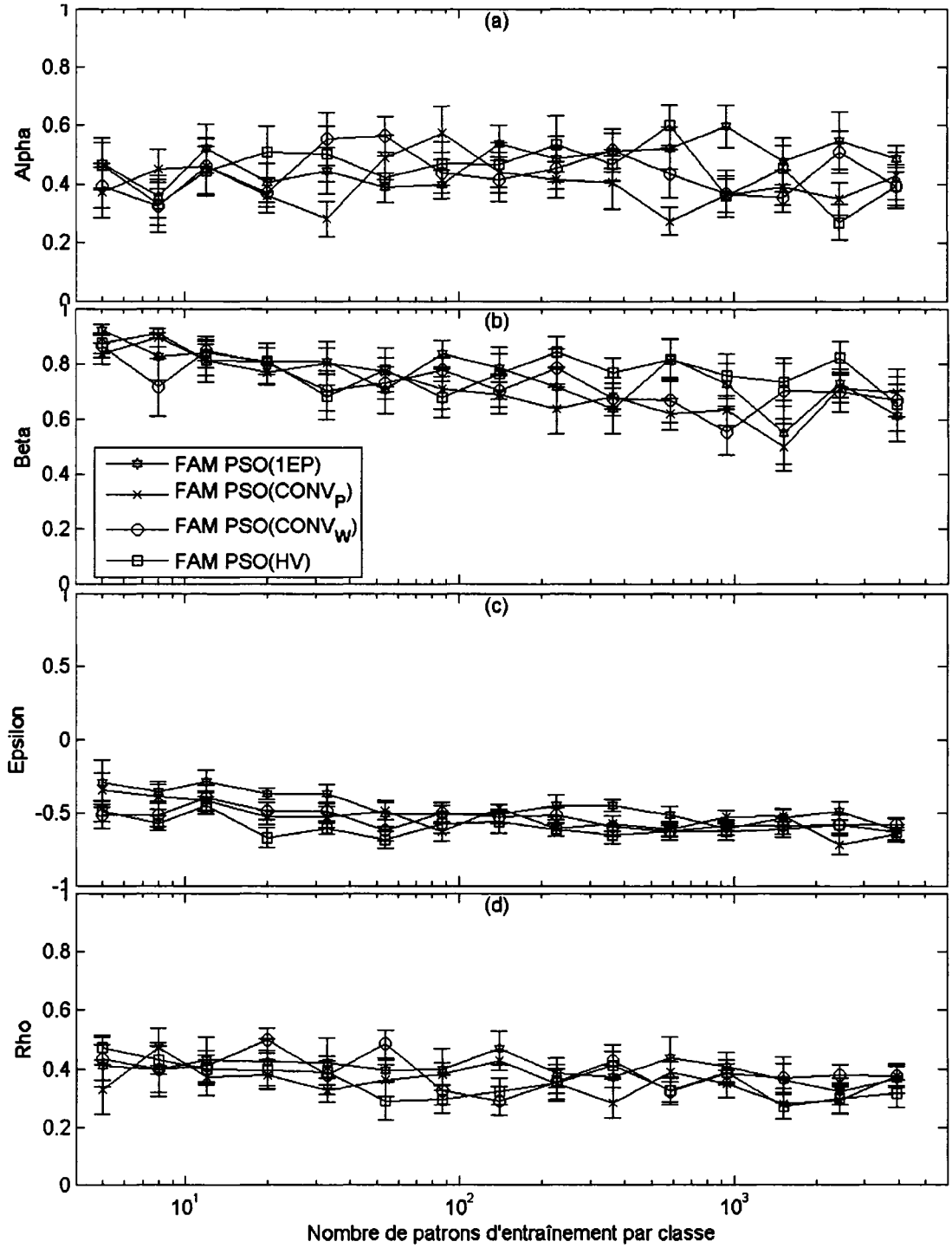


Figure 47 Valeurs des paramètres internes du FAM lors de l'utilisation des stratégies spécialisées PSO avec la base $DB_\sigma(9\%)$

(a) α: paramètre de choix, (b) β : vitesse d'apprentissage, (c) ε : paramètre de MatchTracking et (d) $\bar{\rho}$: vigilance de base

4.1.2 Analyse

Tel que le montrent les figures 39 à 46, il est clair que l'optimisation des paramètres internes des réseaux fuzzy ARTMAP est très avantageux au point de vue des erreurs en généralisation et des taux de compression.

L'optimisation des paramètres permet d'obtenir des erreurs en généralisation très proches des limites théoriques et ce, pour toutes les tailles des bases d'apprentissage ainsi que pour toutes les stratégies d'apprentissage testées avec les bases de données possédant du chevauchement. Ainsi, même avec la plus petite taille de base d'apprentissage, les réseaux créés à l'aide de nos stratégies d'apprentissage spécialisées obtiennent des performances proches de l'erreur théorique et ce, pour tous les degrés de chevauchement testés. De plus, les erreurs en généralisation réalisées par les réseaux fuzzy ARTMAP sont généralement plus petites que celles obtenues avec le classificateur *k*NN et sont comparables à celles obtenues avec CQB.

Au chapitre 3 , lors de l'optimisation de la taille de la base d'apprentissage sur un des tests de la base $DB_{\mu}(9\%)$, nous avons réussi à diminuer l'erreur en généralisation à 9.87% (MT+ et HV) alors qu'elle était de 14.85% avec la taille maximale de la base d'entraînement. En optimisant les paramètres avec cette même base de données nous obtenons une erreur en généralisation de 9.07% avec la taille maximale et lors de l'optimisation de la taille de la base d'entraînement nous obtenons une erreur de 8.99% avec 1519 patrons par classe[2]. Afin de mieux comprendre l'impact de l'optimisation des paramètres, la Figure 48 présente les bornes de décision de ces tests pour $DB_{\mu}(9\%)$, avec

[2]Certaines erreurs en généralisation obtenues sont juste inférieures à l'erreur théorique. Ce phénomène est provoqué par le manque de patrons dans les bases de données. Ainsi, le nombre de patrons contenus dans la base de test n'est pas suffisant pour représenter parfaitement la distribution calculée par les coefficients du Tableau III et du Tableau IV. Les erreurs réelles de ces bases de données sont donc très proches des erreurs théoriques, mais elles ne sont pas identiques.

la taille de la base d'entraînement optimisée et maximale, lors de l'optimisation des paramètres ainsi que lors de l'utilisation des paramètres standard (MT+).

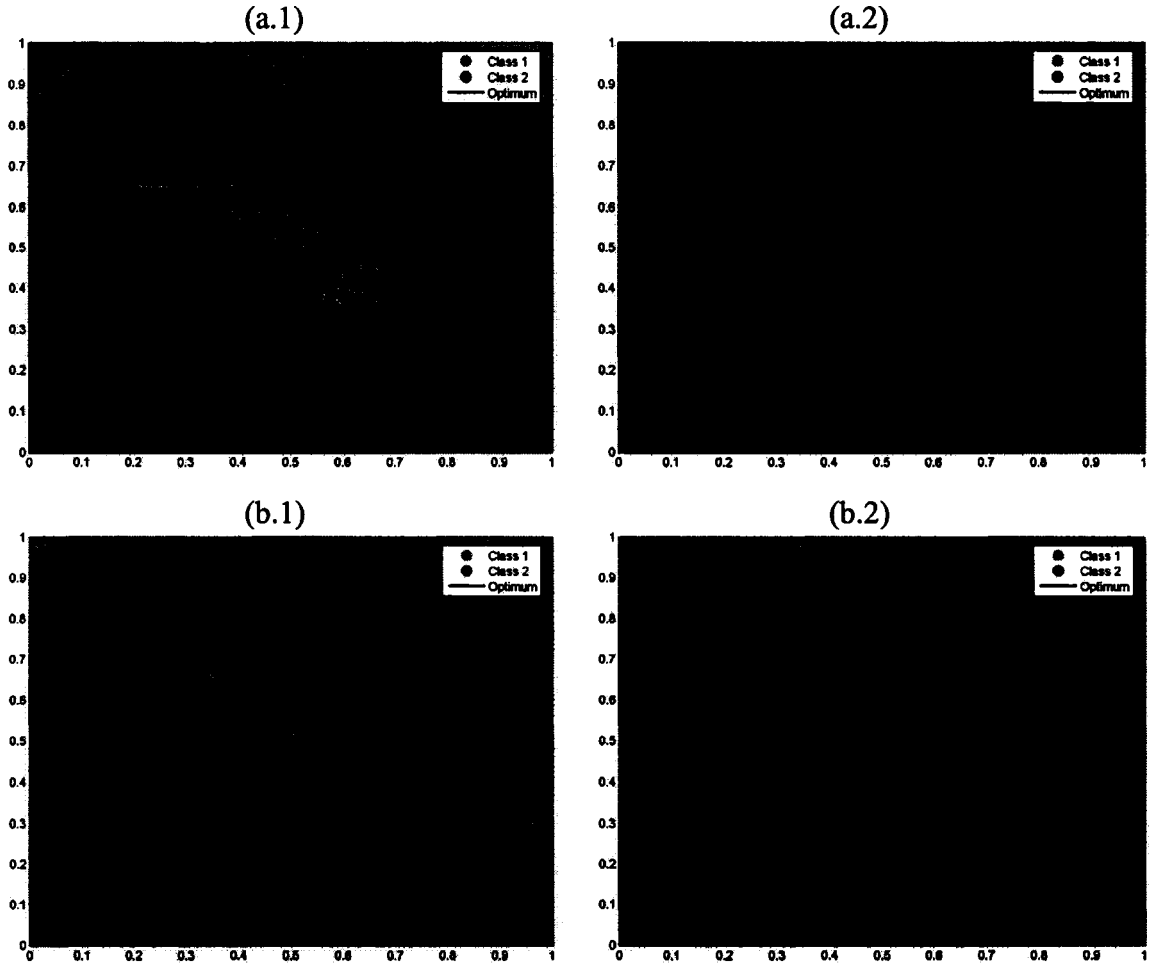


Figure 48 Bornes obtenues avec et sans optimisation des paramètres ainsi qu'avec et sans optimisation de la taille de la base d'apprentissage pour $DB_{\mu}(9\%)$

Soit (a) sans optimisation des paramètres et (b) avec optimisation des paramètres, (1) avec la taille maximale de la base d'entraînement (5000 patrons par classe) et (2) avec la taille de la base d'apprentissage optimisée (a.2 = 178 patrons par classe et b.2 = 1519 patrons par classe); le tout avec la stratégie d'apprentissage HV.

Le tableau X présente le sommaire des résultats des quatre réseaux fuzzy ARTMAP présentés à la figure 48.

Tableau X

Résultats avec et sans optimisation des paramètres avec HV pour une série de $DB_{\mu}(9\%)$

HV	Nombre de patrons d'entraînement par classe	Erreur en généralisation	Nombre de catégories	Taux de compression
Sans PSO	5000	14.85%	916	5.46
Sans PSO	178	9.87%	4	44.5
Avec PSO	5000	9.07%	2	2500
Avec PSO	1519	8.99%	2	759.5

Les réseaux obtenus avec l'optimisation des paramètres possèdent, en moyenne, une catégorie par classe. En sachant que les classes des bases de données DB_{μ} et DB_{σ} sont des distributions gaussiennes et que les performances en généralisation obtenues sont très proches des erreurs théoriques, les catégories créées par ces réseaux représentent donc les centres de masse des distributions gaussiennes.

On remarque également que les quatre stratégies d'apprentissage offrent des performances équivalentes au niveau des erreurs en généralisation ainsi qu'au niveau des taux de compression. Par contre, le temps de convergence de la stratégie PSO(1EP) est inférieur aux trois autres stratégies d'apprentissage spécialisées. Ainsi, bien que l'optimisation des paramètres requière toujours des temps de convergence beaucoup plus élevés que sans optimisation, nous sommes en mesure de réduire ce coût en utilisant la stratégie d'apprentissage PSO(1EP) sans dégrader les performances en généralisation ainsi que les taux de compression.

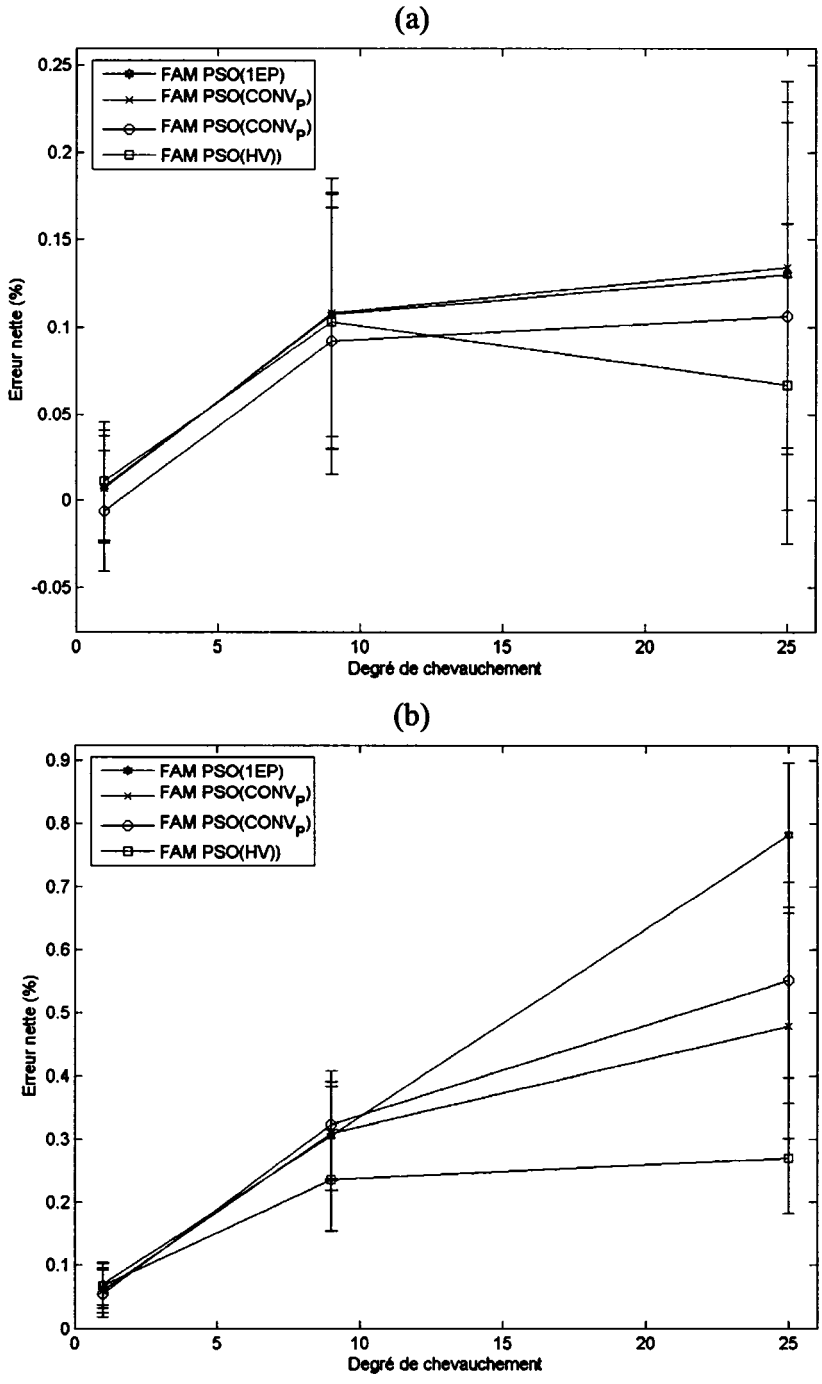


Figure 49 Erreur nette avec PSO en fonction du chevauchement avec DB_μ

(a) avec la taille de la base d'apprentissage optimisée et (b) avec la taille maximale de la base d'apprentissage

Pour appuyer cette analyse, la figure 49 présente l'erreur nette obtenue lors de l'utilisation de la taille maximale de la base d'apprentissage et lors de l'optimisation du nombre de patrons d'entraînement. Tel qu'on peut le voir, avec la taille maximale de la base d'entraînement (figure 49(b)) la stratégie d'apprentissage PSO(1EP) présente des erreurs en généralisation supérieures aux autres stratégies avec 25% de chevauchement. Malgré ce fait, une fois l'optimisation du nombre de patrons d'entraînement effectuée (Figure 49(a)), toutes les stratégies obtiennent des erreurs en généralisation similaires.

Pour montrer que la stratégie PSO(1EP) obtient des performances en généralisation similaires à PSO(HV), les figures 49, 50 et 51 présentent les différences des résultats obtenus pour ces deux stratégies, et ce, pour l'erreur en généralisation, le taux de compression, le temps de convergence et le nombre de catégories créées avec, respectivement, les bases $DB_{\mu}(1\%)$, $DB_{\mu}(9\%)$ et $DB_{\mu}(25\%)$. Ainsi, lorsque la courbe est positive, la valeur obtenue avec la stratégie PSO(HV) est plus grande que celle obtenue avec la stratégie PSO(1EP), et vice-versa.

On remarque avec ces deux figures que les erreurs en généralisation avec la stratégie PSO(1EP) sont similaires de celles obtenues avec PSO(HV). Les taux de compression obtenus avec la stratégie PSO(1EP) sont similaires ou meilleurs que ceux obtenus avec PSO(HV). De ce fait, le nombre de catégories créées par PSO(1EP) est généralement égal ou plus petit que celui obtenu avec PSO(HV). Le temps de convergence est nettement plus petit avec PSO(1EP). En utilisant PSO(1EP) plutôt que PSO(HV), on épargne, en moyenne, aux alentours de $3x10^4$ époques d'entraînement.

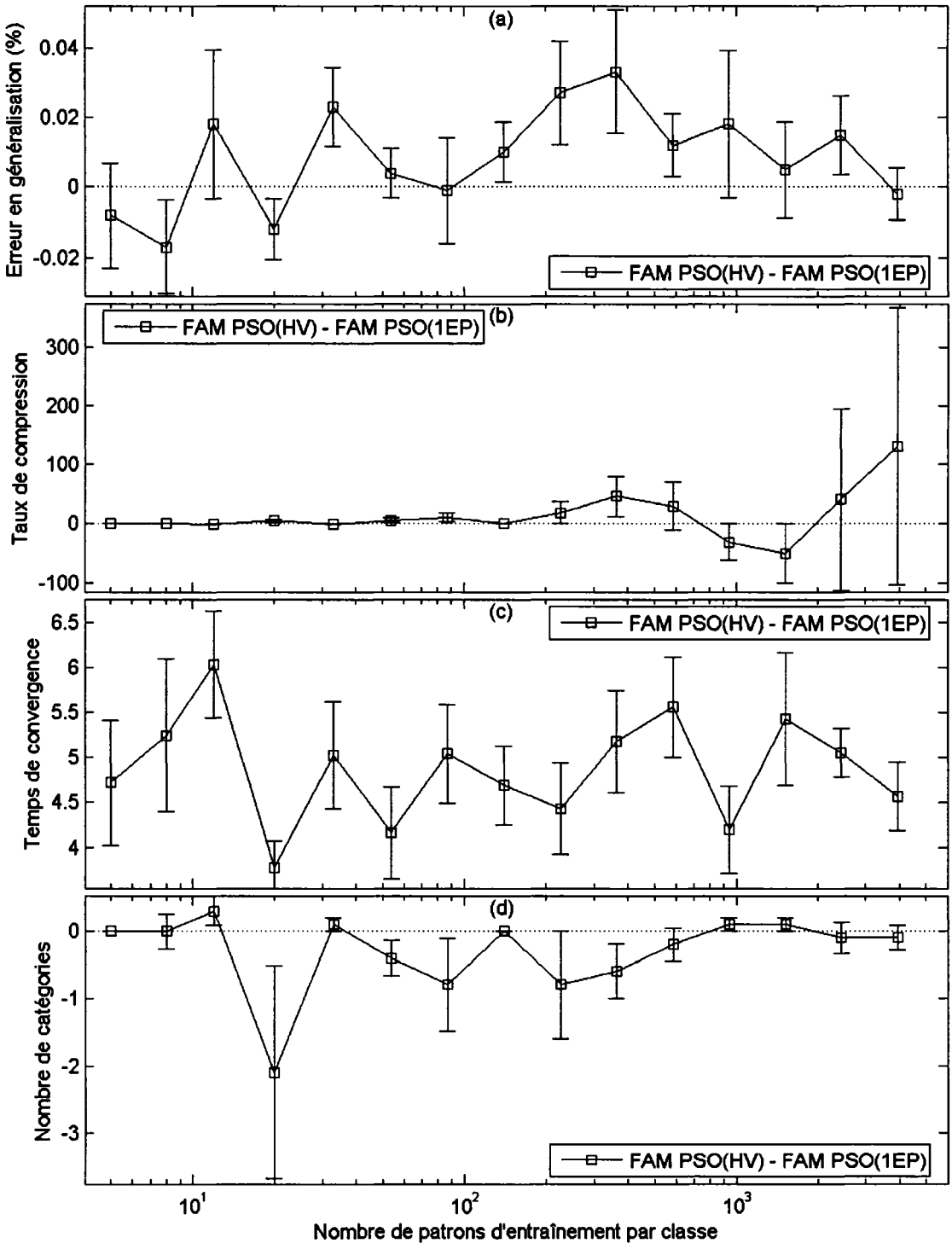


Figure 50 Différence entre PSO(HV) et PSO(1EP) avec la base $DB_{\mu}(1\%)$

(a) Erreur en généralisation, (b) Taux de compression, (c) Temps de convergence et (d) Nombre de catégories.

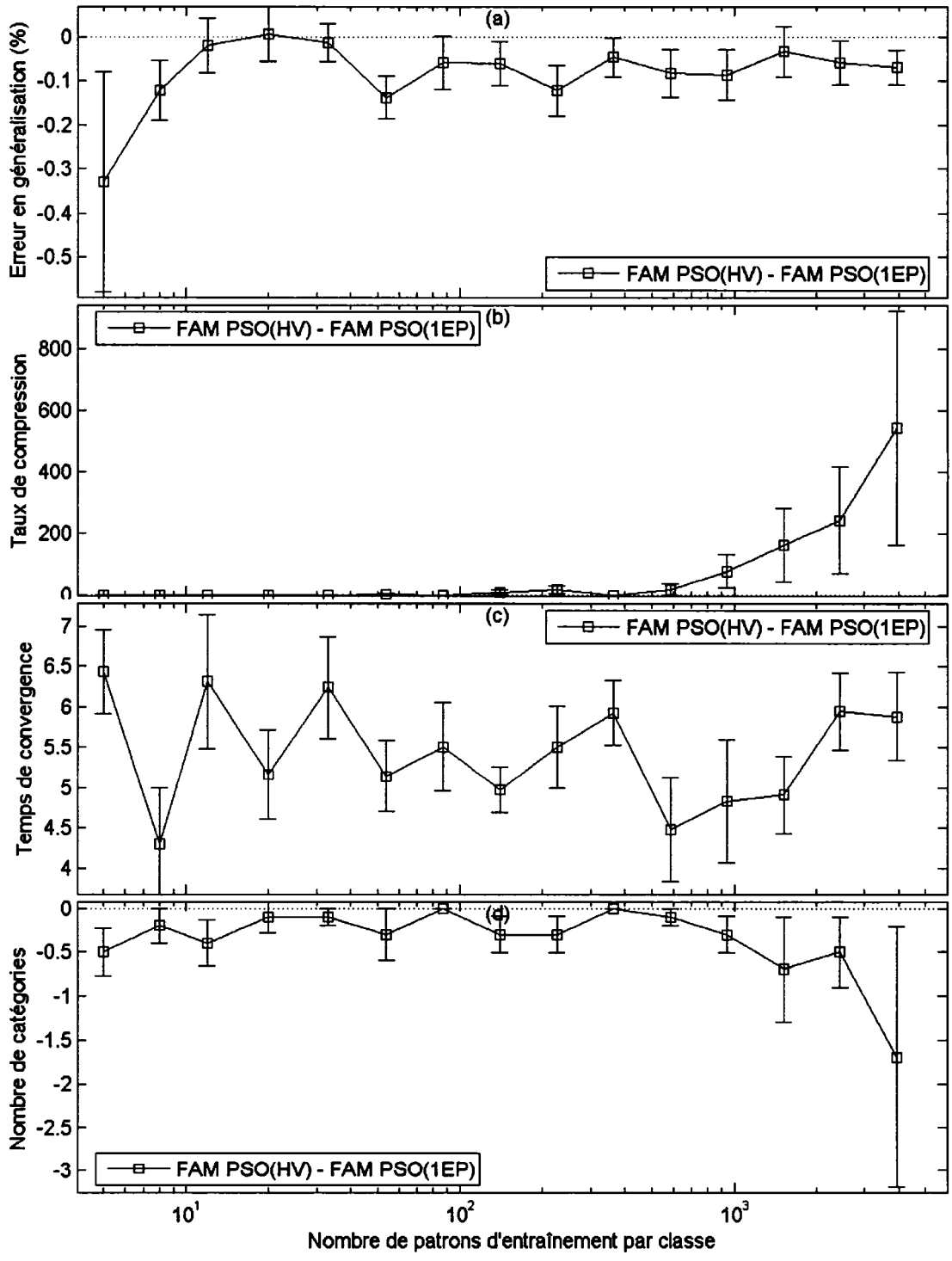


Figure 51 Différence entre PSO(HV) et PSO(1EP) avec la base $DB_{\mu}(9\%)$

(a) Erreur de généralisation, (b) Taux de compression, (c) Temps de convergence et (d) Nombre de catégories.

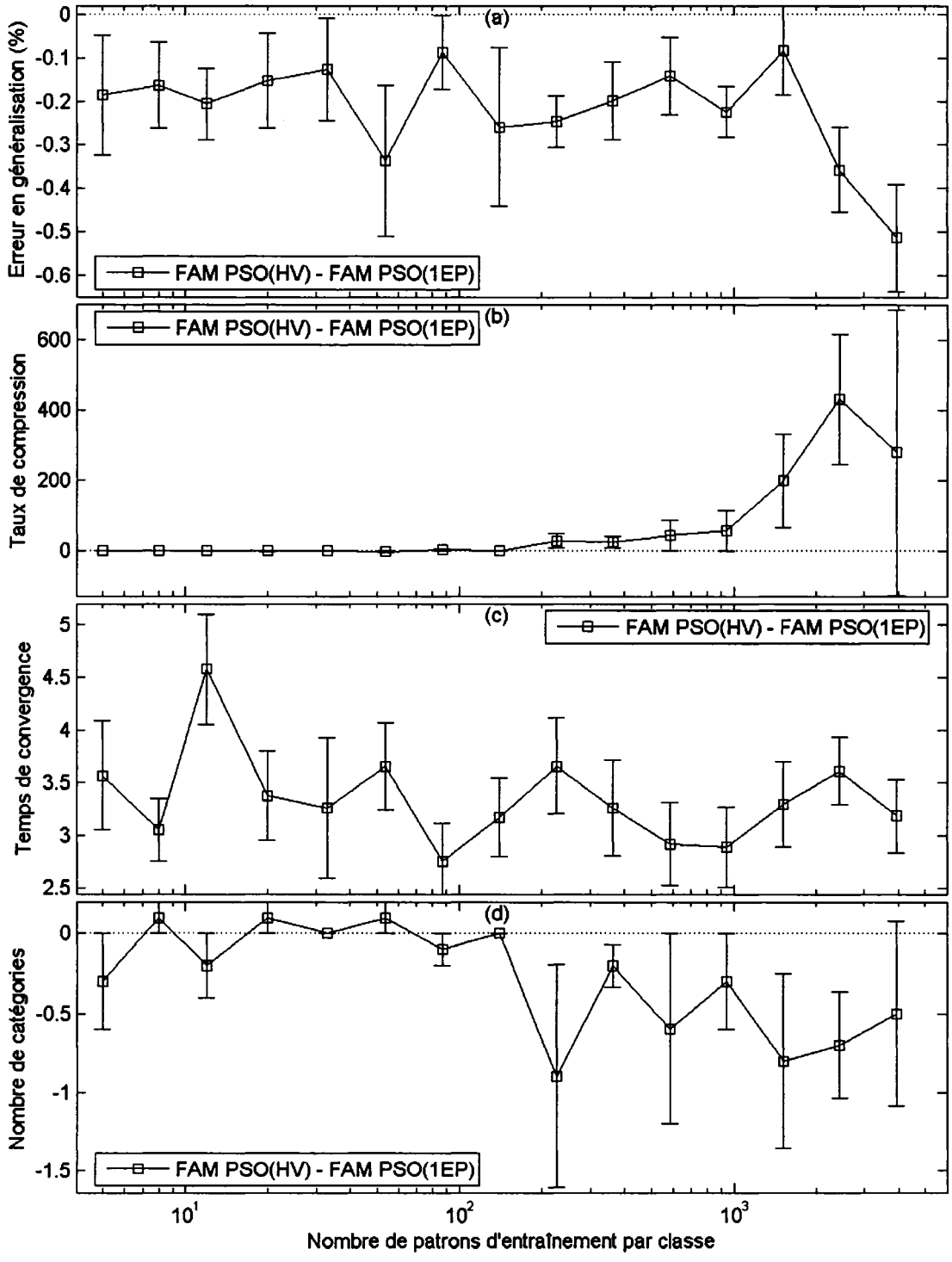


Figure 52 Différence entre PSO(HV) et PSO(1EP) avec la base $DB_{\mu}(25\%)$

(a) Erreur en généralisation, (b) Taux de compression, (c) Temps de convergence et (d) Nombre de catégories.

La Figure 49(a) démontre également qu'une fois les paramètres optimisés, il reste encore une légère dégradation des performances en généralisation engendrée par la taille de la base d'entraînement. Pour mieux voir ce phénomène, la Figure 53 présente l'erreur de sur-apprentissage due à la taille de la base d'entraînement lors de l'utilisation des stratégies d'apprentissage spécialisées pour FAM avec les bases DB_{μ}.

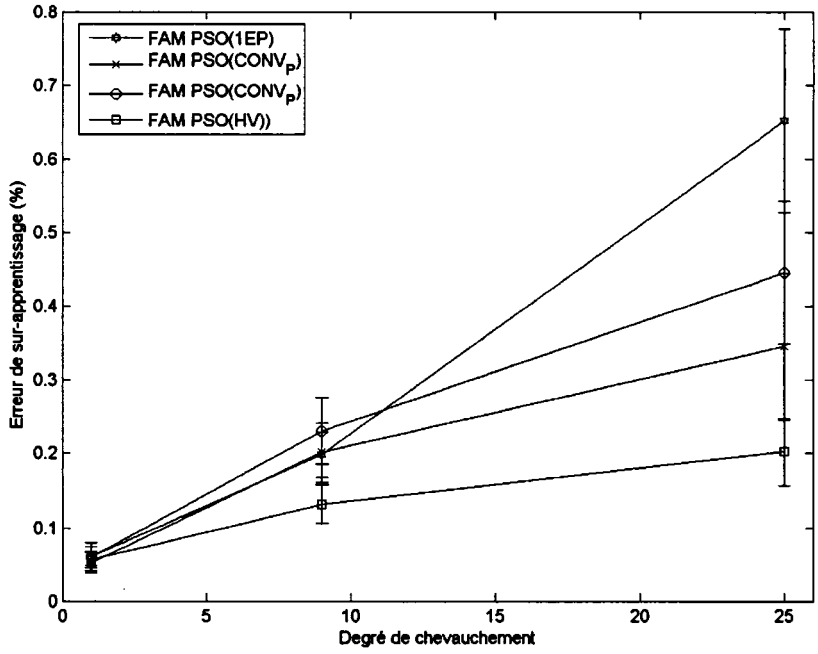


Figure 53 Erreur de sur-apprentissage due à la taille de la base d'entraînement avec les stratégies d'apprentissage spécialisées pour DB_{μ}

En tenant compte de ces résultats, nous venons de démontrer que, contrairement à ce qu'il a été énoncé dans la conclusion du chapitre 3, le facteur causant la plus grande dégradation des performances à l'intérieur des réseaux FAM n'est pas la taille de la base d'apprentissage. Il faut plutôt dire que, lors de l'utilisation des paramètres standard MT+, la taille de la base d'apprentissage est le facteur le plus influent dans la création d'erreur de sur-apprentissage. Par contre, lors de l'optimisation des paramètres des réseaux FAM avec PSO, nous avons remarqué, qu'en fait, ce sont les paramètres internes de ces réseaux qui ont le plus grand impact sur la dégradation des performances.

Ainsi, la cause première des mauvaises performances des réseaux fuzzy ARTMAP est la sélection des valeurs des paramètres utilisés lors de la phase d'entraînement.

On remarque également sur les figures 39 à 46 que le taux de compression augmente de façon linéaire. Ceci indique que le nombre de catégories créées avec les stratégies PSO reste constant pour toutes les trailles des bases d'apprentissage, soit autour de 2.2 en moyenne. Ce faible nombre de catégories est dû aux valeurs attribuées aux paramètres lors de l'optimisation. Ces valeurs permettent ainsi d'éviter la création de catégories qui n'amélioreront pas les performances en généralisation et ainsi obtenir un réseau compact, simple, dont les performances sont très proches de l'erreur théorique.

On remarque également que les valeurs utilisées pour le paramètre epsilon sont toutes négatives. Ceci montre, tel qu'énoncé au chapitre 3, que le MatchTracking négatif était la bonne direction à suivre pour les bases de données possédant du chevauchement, mais que la valeur absolue de ce paramètre n'était pas assez grande. Les valeurs d'epsilon obtenues oscillent généralement entre -0,2 et -0,7 comparativement à -0,001 lors de l'utilisation des paramètres par défaut MT-. Une grande valeur positive d'epsilon favorise la création de nouvelles catégories, lors de la phase d'apprentissage, lorsque la première catégorie sélectionnée pour classifier un patron d'entraînement n'appartient pas à la bonne classe. À l'opposé, une grande valeur négative de ce paramètre favorise la réutilisation des catégories déjà existantes et ainsi évite la création de nouvelles catégories.

La vigilance de base $\overline{\rho}$ oscille en moyenne entre 0.5 et 0.9, et ce pour tous les degrés de chevauchement testés. Ce paramètre influence directement le nombre de catégories créées lors de la phase d'apprentissage. Une grande valeur de ce paramètre réduit la probabilité qu'une catégorie soit associée au patron d'entraînement présenté et ainsi favorise l'assignation d'une nouvelle catégorie. À l'opposé, une faible valeur permet à plusieurs catégories d'être sélectionnées comme étant la bonne catégorie pour classifier

le patron d'entraînement et ainsi évite de créer un trop grand nombre de catégories. De ce fait, ce paramètre influence directement la taille maximale des catégories, soit: $|w_j| \leq 2(1-\rho)$.

Le paramètre de choix α oscille en moyenne entre 0.25 et 0.75 avec les bases de données possédant un degré de chevauchement. Ce paramètre influence le nombre d'itérations de recherche des catégories pendant la phase d'apprentissage avant de créer une nouvelle catégorie. Ainsi, une grande valeur de ce paramètre entraîne une recherche rapide à travers les catégories existantes lors de l'entraînement, favorisant la création de nouvelles catégories. Une faible valeur favorise les catégories existantes face à l'assignation d'une nouvelle catégorie, soit une recherche plus approfondie à l'intérieur des catégories existantes.

La vitesse d'apprentissage β oscille en moyenne entre 0.5 et 0.9. Ce paramètre détermine la vitesse à laquelle les catégories s'adaptent aux patrons d'entraînement, chaque fois qu'une catégorie est modifiée. Une grande valeur permet aux catégories de s'adapter le plus rapidement possible alors qu'une faible valeur diminue la vitesse de changement de la taille de la catégorie.
Les valeurs des paramètres obtenues avec les stratégies d'apprentissage spécialisées utilisant l'algorithme d'optimisation PSO favorisent la création d'un petit nombre de catégories (epsilon négatif) possédant une grande surface (vigilance de base élevée).

4.2 Effets de la structure des bases de données

Cette section présente une comparaison entre les deux méthodes de création du degré de chevauchement lors de l'optimisation des paramètres des réseaux fuzzy ARTMAP.

4.2.1 Résultats

Dans la section 3.2, nous avons conclu que les deux structures utilisées pour la création du chevauchement (DB_μ et DB_σ) engendrent des bases caractérisées par un degré de difficulté semblable pour les réseaux fuzzy ARTMAP. Ainsi, les réseaux obtenus avec ces deux méthodes ont des performances analogues sur le plan des erreurs en généralisation, des temps de convergence ainsi que des taux de compression. Cette section montre que ce phénomène est encore présent lors de l'apprentissage avec l'optimisation des paramètres internes des réseaux fuzzy ARTMAP.

Les figures 53, 54 et 55 présentent une comparaison entre la base $DB_\mu(9\%)$ et $DB_\sigma(9\%)$ au niveau des erreurs en généralisation, des temps de convergence ainsi que des taux de compression, pour les quatre stratégies d'apprentissage spécialisées pour FAM. Ainsi, lorsque la courbe est positive, la valeur obtenue avec la base $DB_\mu(9\%)$ est plus grande que celle obtenue avec la base $DB_\sigma(9\%)$, et vice-versa.

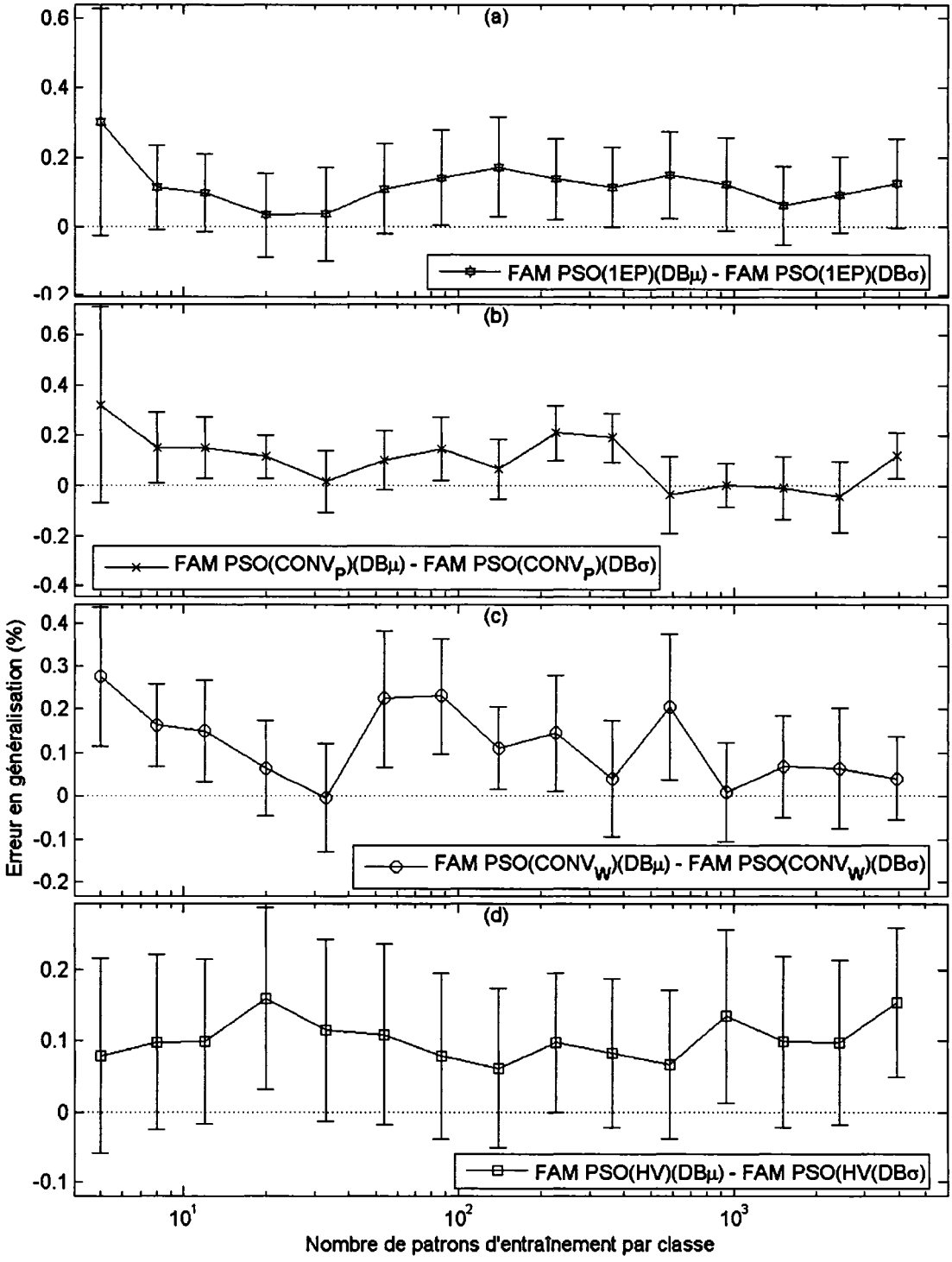


Figure 54 $DB_\mu(9\%)$ versus $DB_\sigma(9\%)$ sur l'erreur en généralisation avec PSO (a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

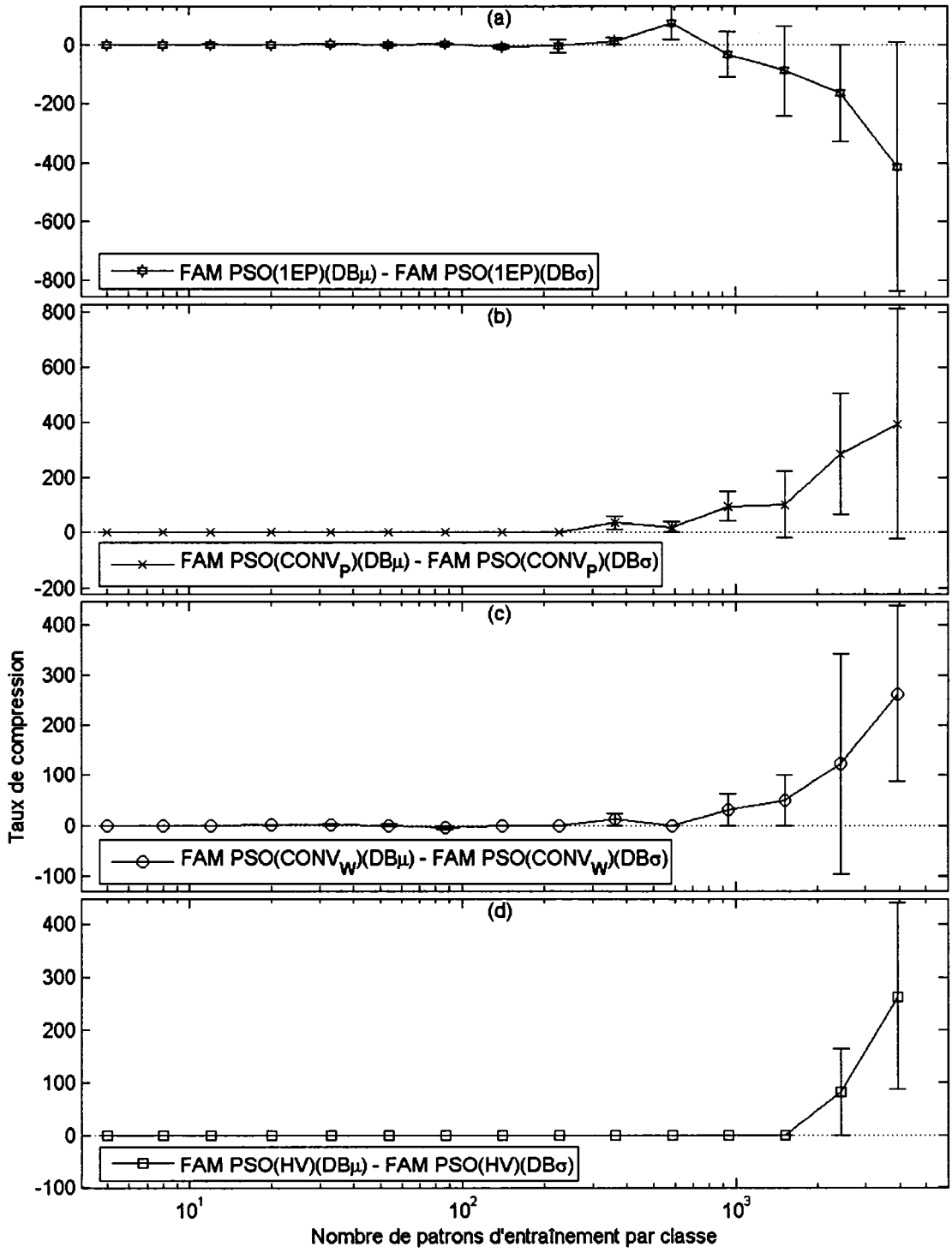


Figure 55 $DB_\mu(9\%)$ versus $DB_\sigma(9\%)$ sur le taux de compression avec PSO
(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

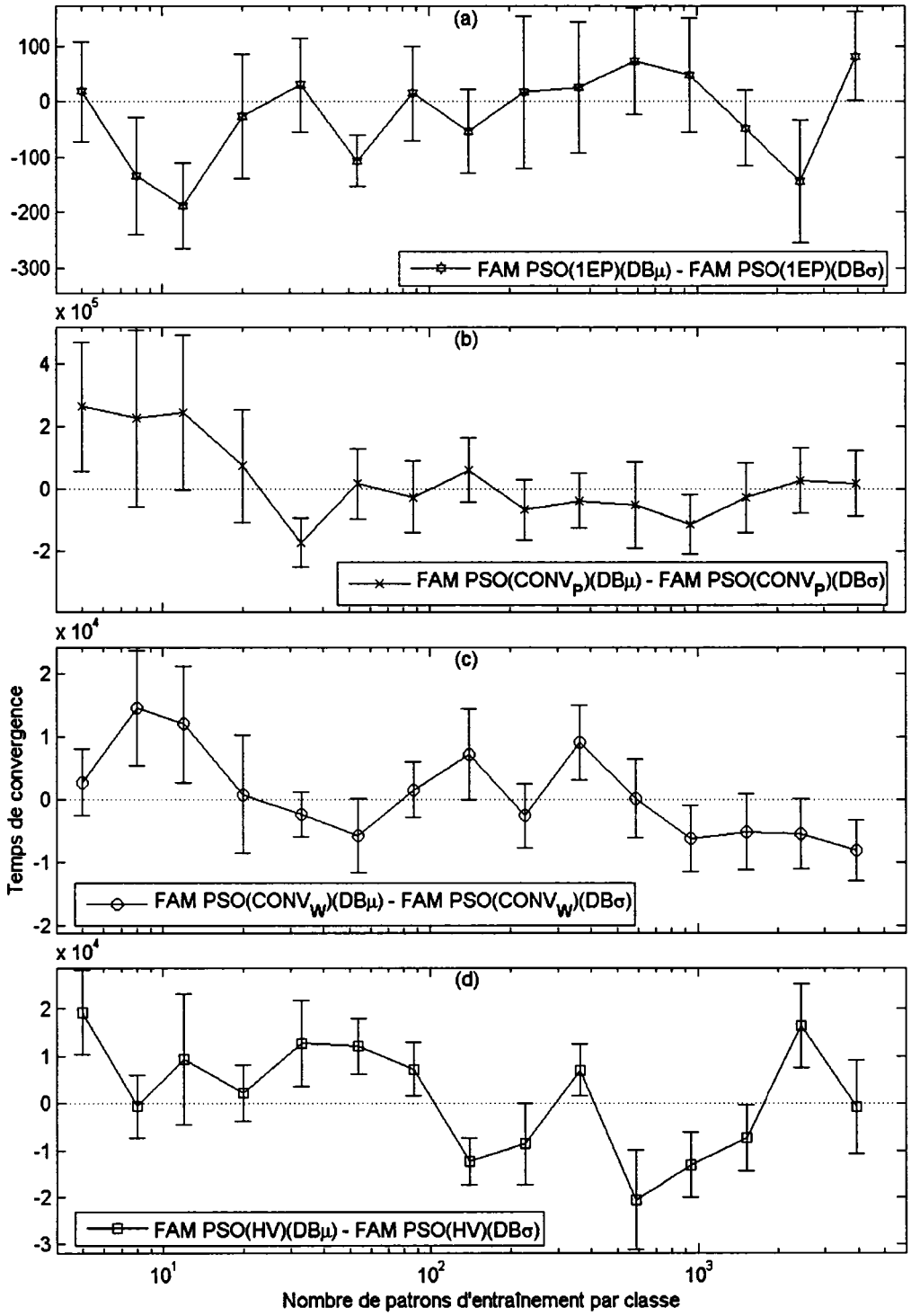


Figure 56 $DB_{\mu}(9\%)$ versus $DB_{\sigma}(9\%)$ sur le temps de convergence avec PSO
(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

4.2.2 Analyse

La Figure 54 montre que les performances en généralisation obtenues avec les bases de données $DB_{\mu}(9\%)$ et $DB_{\sigma}(9\%)$ sont similaires. La dispersion des erreurs en généralisation est calculée avec l'erreur standard (voir équation (2.8)) et puisque les erreurs en généralisation s'étendent sur les deux régions, positives et négatives, ces résultats ne sont pas significativement différents. Ainsi, statistiquement, les deux méthodes de création de chevauchement génèrent des bases caractérisées par un degré de difficulté semblable pour les réseaux fuzzy ARTMAP.

Au niveau des taux de compression, la Figure 55 illustre que, généralement, les résultats obtenus sont similaires. Par contre, lors de l'utilisation de la stratégie d'apprentissage $PSO(CONV_W)$, nous obtenons des taux de compression supérieurs avec la base $DB_{\mu}(9\%)$ lors de l'emploi des grandes tailles de la base d'entraînement.

La Figure 56 présente la comparaison entre la base $DB_{\mu}(9\%)$ et la base $DB_{\sigma}(9\%)$ pour les temps de convergence. Ces résultats sont majoritairement similaires à l'exception de ceux obtenus avec la stratégie d'apprentissage $PSO(CONV_W)$. Cette stratégie demande un plus grand nombre d'époques d'entraînement avec la base $DB_{\sigma}(9\%)$ qu'avec $DB_{\mu}(9\%)$, lors de l'utilisation des grandes tailles de la base d'apprentissage.

Malgré les temps de convergence, la majorité des résultats obtenus par les deux structures des bases de données avec chevauchement sont semblables. Ainsi, nous en tirons la même conclusion qu'au chapitre 3, soit qu'il n'y a pas de différence entre ces deux structures, même lors de l'optimisation des paramètres internes des réseaux fuzzy ARTMAP.

4.3 Bases de données sans chevauchement

Cette sous-section traite des effets de l'optimisation des paramètres internes des réseaux fuzzy ARTMAP avec les stratégies d'apprentissage spécialisées pour les bases de données sans chevauchement (DB_{P2} et DB_{CiS}). Étant donné qu'aucun effet dû à la normalisation n'est présent avec les paramètres standard FAM (voir section 3.3), la technique de normalisation MinMax est utilisée.

4.3.1 Résultats

Les figures suivantes présentent les erreurs en généralisation, les temps de convergence et les taux de compression obtenus pour les quatre stratégies d'apprentissage spécialisées avec les bases de données DB_{CiS} et DB_{P2}. De plus, les valeurs optimisées des quatre paramètres internes des réseaux fuzzy ARTMAP sont présentées. À noter que le temps de convergence inclut toutes les époques effectuées par toutes les particules lors de l'optimisation PSO.

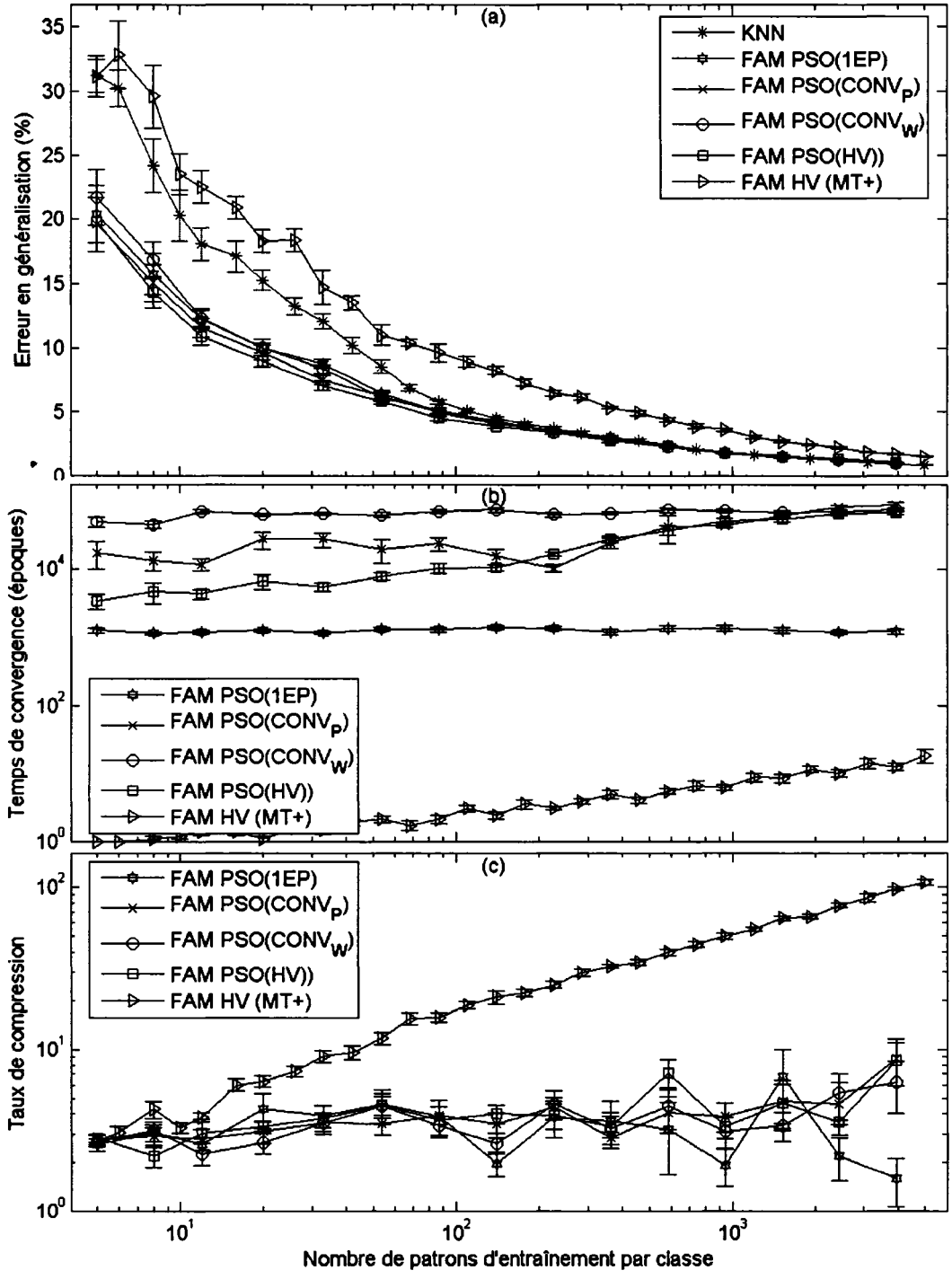


Figure 57 Performance du FAM avec les stratégies PSO sur la base DB_{CIS}

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

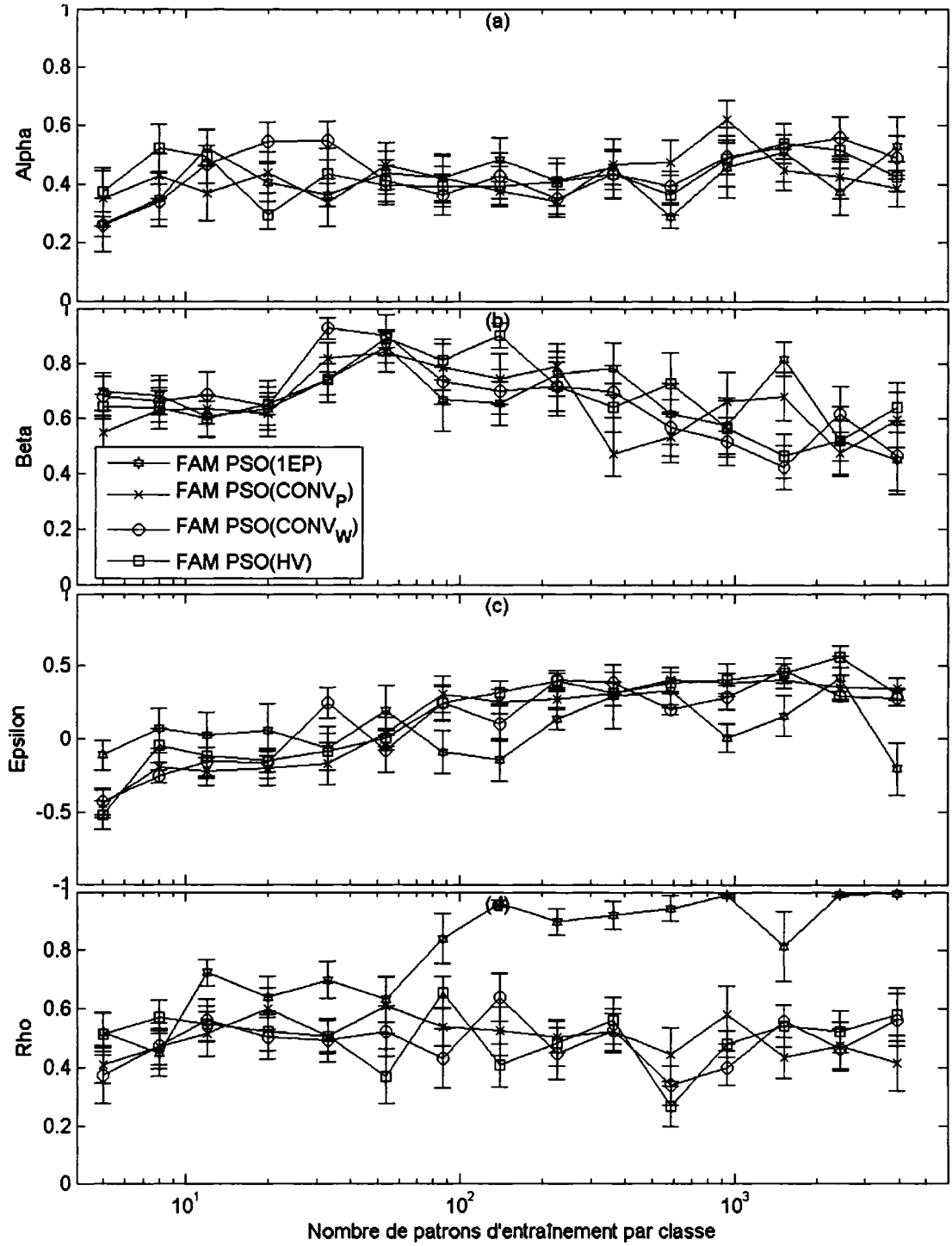


Figure 58 Valeurs des paramètres internes du FAM lors de l'utilisation des stratégies spécialisées PSO avec la base DB_{CIS}

(a) α: paramètre de choix, (b) β : vitesse d'apprentissage, (c) ε : paramètre de MatchTracking et (d) $\overline{\rho}$: vigilance de base

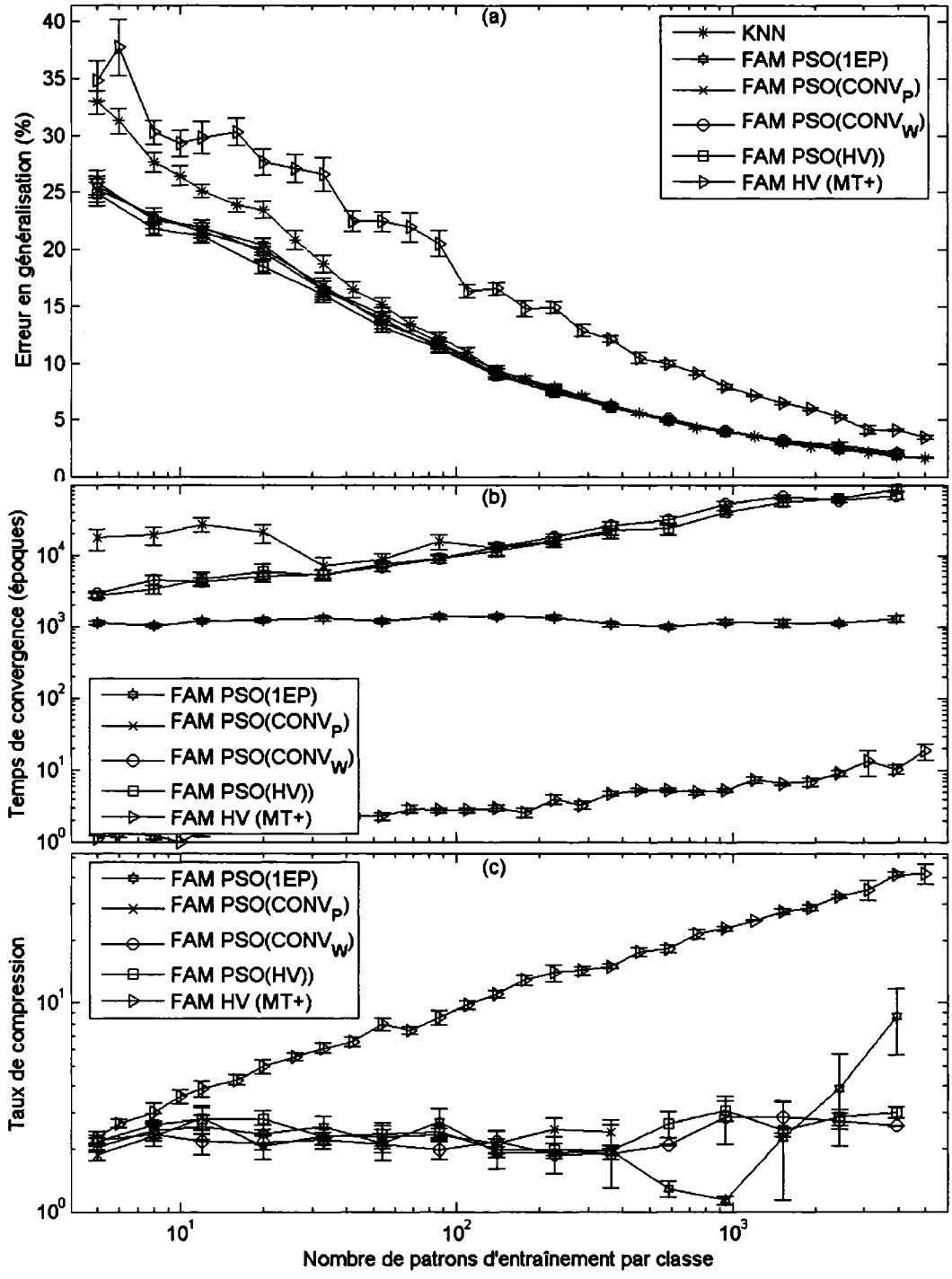


Figure 59 Performance du FAM avec les stratégies PSO sur la base DB_{P2}

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

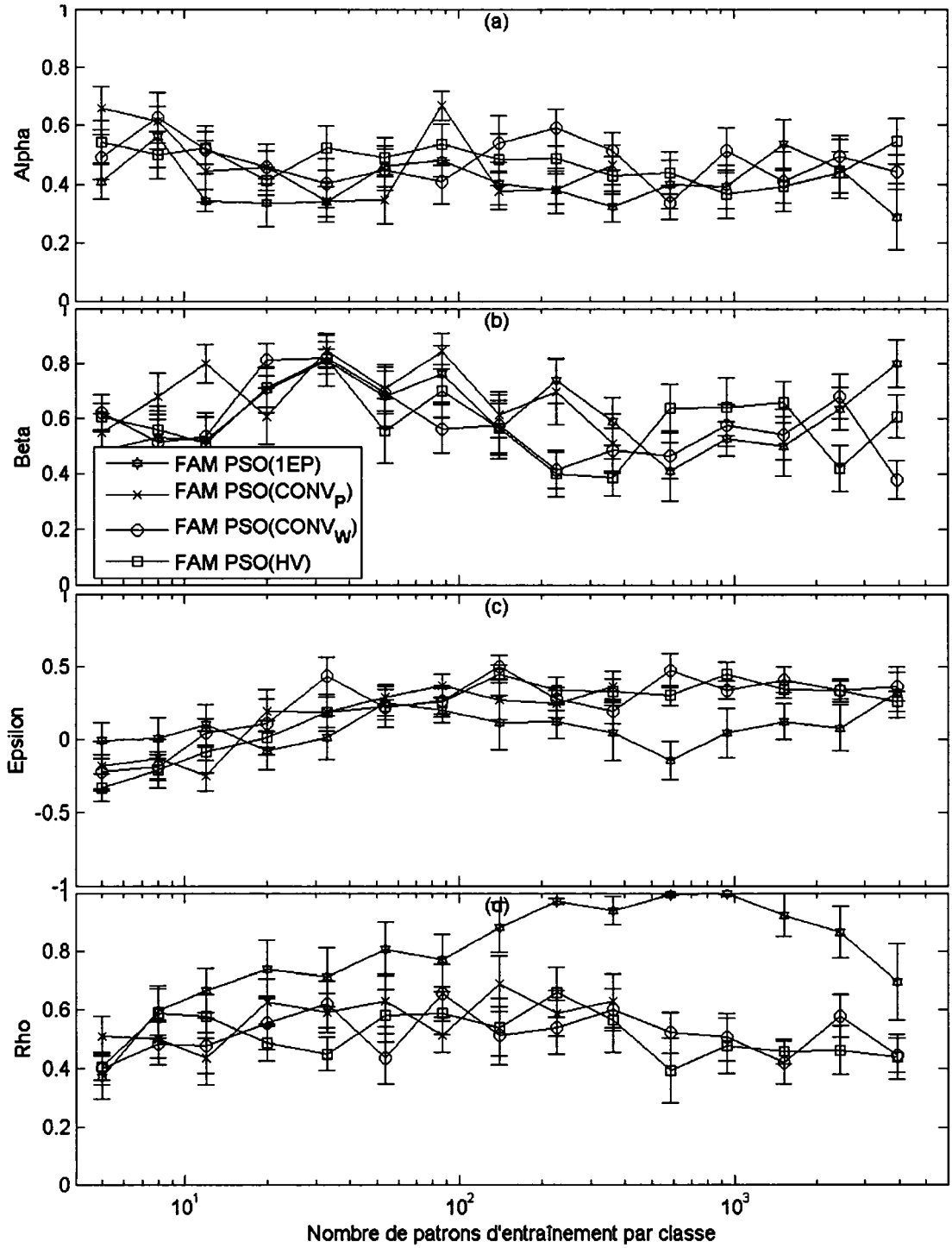


Figure 60 Valeurs des paramètres internes du FAM lors de l'utilisation des stratégies spécialisées PSO avec la base DB_{P2}

(a) α: paramètre de choix, (b) β : vitesse d'apprentissage, (c) ε : paramètre de MatchTracking et (d) $\overline{\rho}$: vigilance de base

4.3.2 Analyse

Les figures précédentes montrent que les stratégies d'entraînement spécialisées, effectuant l'optimisation des paramètres internes des réseaux fuzzy ARTMAP, permettent de meilleures performances en généralisation, particulièrement avec les bases d'apprentissage de petites tailles.

L'optimisation de ces paramètres permet également de générer des performances en généralisation comparables à celles obtenues avec le *k*NN et ce, pour les quatre stratégies d'apprentissage utilisées. À noter que lors de l'utilisation de *k*NN avec les bases de données sans chevauchement, les valeurs de *k* sélectionnées sont presque toujours égales à 1. Avec un faible nombre de patrons d'apprentissage, les erreurs en généralisation relevées sont moindres que les celles obtenues avec le *k*NN. Par contre, plus la taille de la base d'entraînement augmente, plus cet écart diminue, pour finalement disparaître.

On remarque également que les performances en généralisation des quatre stratégies d'apprentissage sont similaires. Puisque la stratégie d'entraînement PSO(1EP) obtient les temps de convergence les plus rapides, il serait préférable de l'utiliser lors de l'optimisation des paramètres, afin de minimiser le temps de convergence avec les bases de données sans chevauchement et ce, sans dégradation des performances en généralisation. Au niveau des taux de compression, on remarque des résultats similaires entre les quatre stratégies d'apprentissage, sauf lorsque la taille de la base d'entraînement est très grande. Ainsi, avec le nombre maximal de patrons d'apprentissage, les taux de compression obtenus avec la stratégie PSO(1EP) sont différents de ceux obtenus avec les trois autres stratégies. Avec la base DB_{CiS}, PSO(1EP) obtient des taux plus faibles que les trois autres stratégies, alors qu'avec DB_{P2} elle obtient les meilleurs taux. Malgré ce phénomène, nous recommandons quand même d'utiliser la stratégie d'apprentissage PSO(1EP) avec l'optimisation des paramètres.

Pour appuyer cette analyse, les figures 60 et 61 présentent la comparaison entre les résultats obtenus pour les stratégies PSO(HV) et PSO(1EP), et ce pour l'erreur en généralisation, le taux de compression, le temps de convergence et le nombre de catégories créées avec les bases DB_{CiS} et DB_{P2}. Ainsi, lorsque la courbe est positive, la valeur obtenue avec la stratégie PSO(HV) est plus grande que celle obtenue avec la stratégie PSO(1EP), et vice-versa.

On observe sur ces deux figures que, lors de l'utilisation de la taille maximale de la base d'apprentissage, les différences entre les stratégies PSO(HV) et PSO(1EP) sont pratiquement nulles. Les taux de compression de la base DB_{CiS} favorisent la stratégie PSO(HV) alors que ceux obtenus avec DB_{P2} favorisent PSO(1EP). Le grand avantage d'utiliser la stratégie d'apprentissage PSO(1EP) se situe au niveau des temps de convergence. Avec la taille maximale de la base d'apprentissage, les temps de convergence obtenus avec la stratégie PSO(1EP) sont de $7x10^4$ à $9x10^4$ époques d'entraînement plus rapides qu'avec la stratégie PSO(HV).

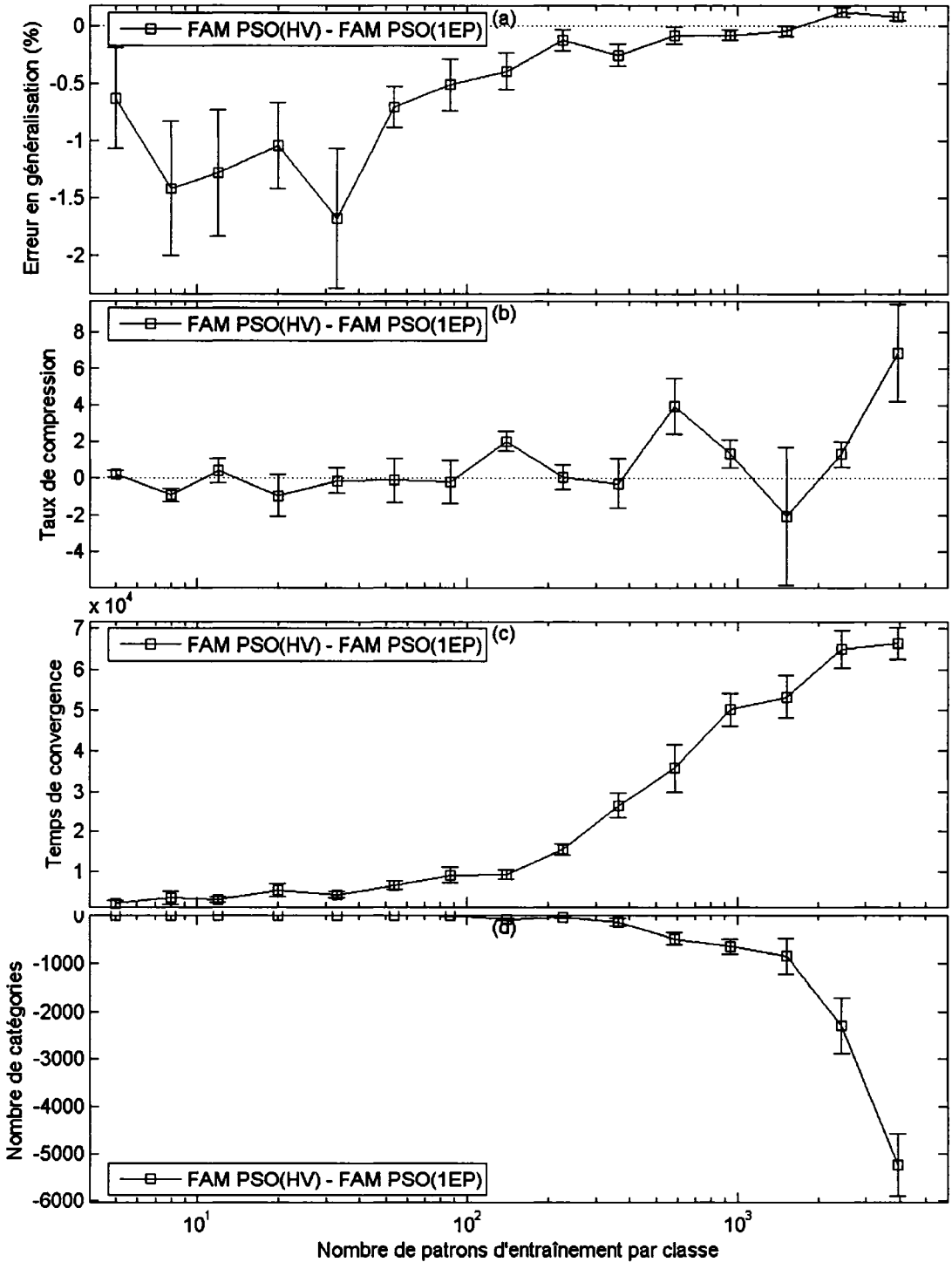


Figure 61 Différence entre PSO(HV) et PSO(1EP) avec la base DB_{Cis}

(a) Erreur en généralisation, (b) Taux de compression, (c) Temps de convergence et (d) Nombre de catégories.

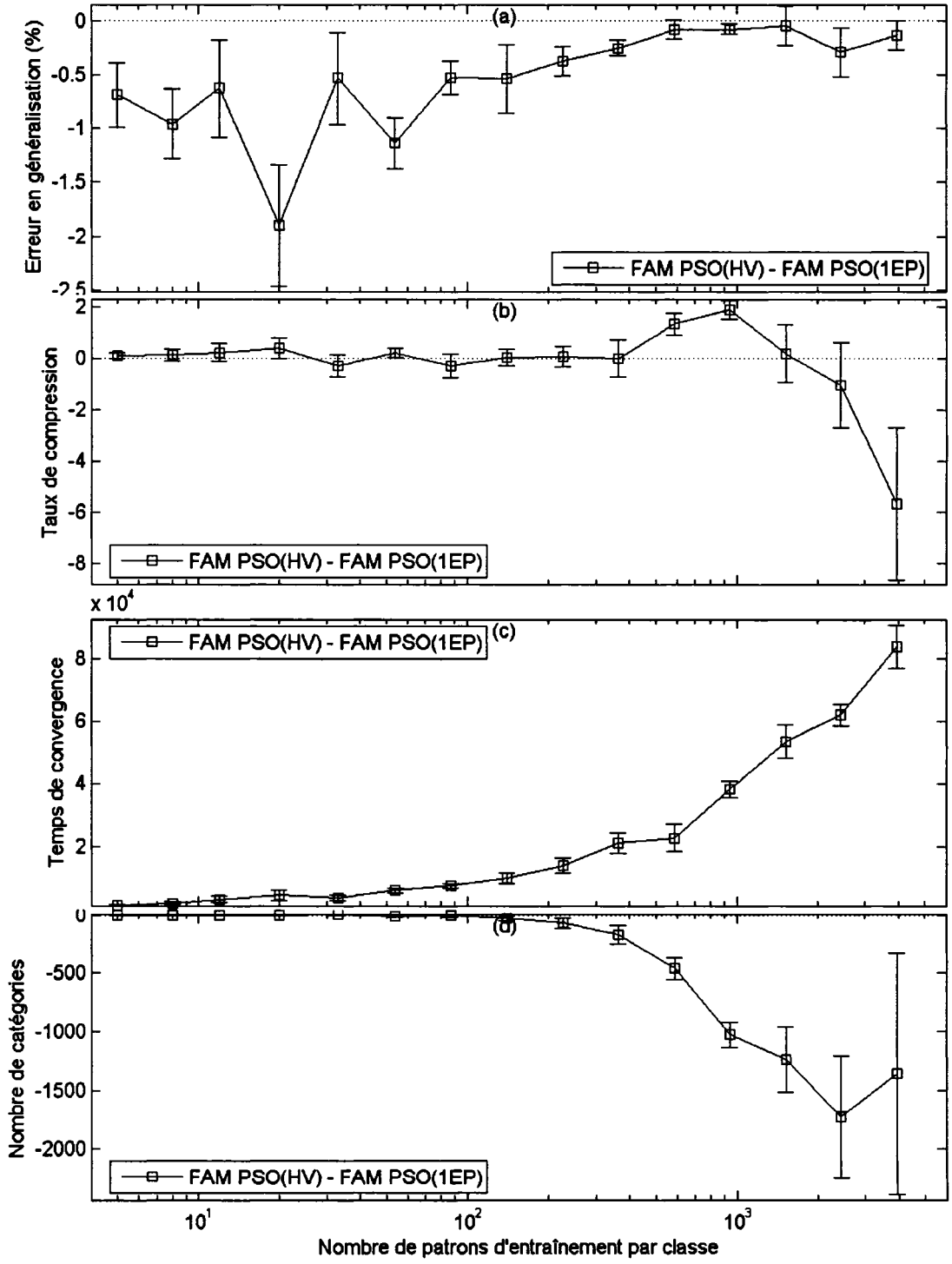


Figure 62 Différence entre PSO(HV) et PSO(1EP) avec la base DB_{P2}

(a) Erreur en généralisation, (b) Taux de compression, (c) Temps de convergence et (d) Nombre de catégories.

Sans l'optimisation des paramètres, il n'y a pas de dégradation des performances en généralisation causée par la taille de la base d'entraînement avec les bases de données sans chevauchement. Ceci est encore le cas lorsque les paramètres sont optimisés par les stratégies d'apprentissage spécialisées. Au chapitre 3, la Figure 15 présente les bornes de décision ainsi que les catégories créées lors de l'augmentation progressive de la taille de la base de données d'entraînement pour une des répétitions avec la base DB_{CiS}. Cette augmentation fait passer l'erreur en généralisation de 19.6% (12 patrons par classe), à 5.59% (226 patrons par classe) pour atteindre 1.66% (5000 patrons par classe), et ce pour la stratégie d'apprentissage HV. Lors de l'optimisation des paramètres, nous obtenons, pour cette même expérience, les erreurs en généralisation suivantes : 15.58% (12 patrons par classe), 3.51% (226 patrons par classe) et 1.36% (5000 patrons par classe) avec la stratégie PSO(HV). Pour mieux comprendre le gain des performances en généralisation obtenu lors de l'utilisation de la stratégie PSO(HV) avec la base BD_{CiS}, la Figure 63 présente les bornes de décision pour ce cas.

La Figure 63 montre que les bornes de décision obtenues avec l'optimisation des paramètres sont plus nettes et suivent mieux la borne optimale qu'avec l'utilisation de paramètres standard MT+.

La Figure 58 nous donne également un indicatif sur les valeurs optimisées des paramètres lors de l'application de l'algorithme PSO avec la base DB_{CiS}. Ainsi, on remarque que la valeur d'epsilon a tendance à être positive lors de l'utilisation d'un grand nombre de patrons d'apprentissage. Le paramètre $\overline{\rho}$ tend vers des valeurs positives aux alentours de 0,5. Pour mieux voir l'effet de ces paramètres et comprendre comment ceux-ci peuvent obtenir une meilleure borne de décision, la Figure 64 présente une comparaison entre les catégories obtenues avec et sans optimisation lors de l'utilisation de la taille maximale de la base d'apprentissage (5000 patrons par classe) sur une des 10 répétitions avec la base DB_{CiS}.

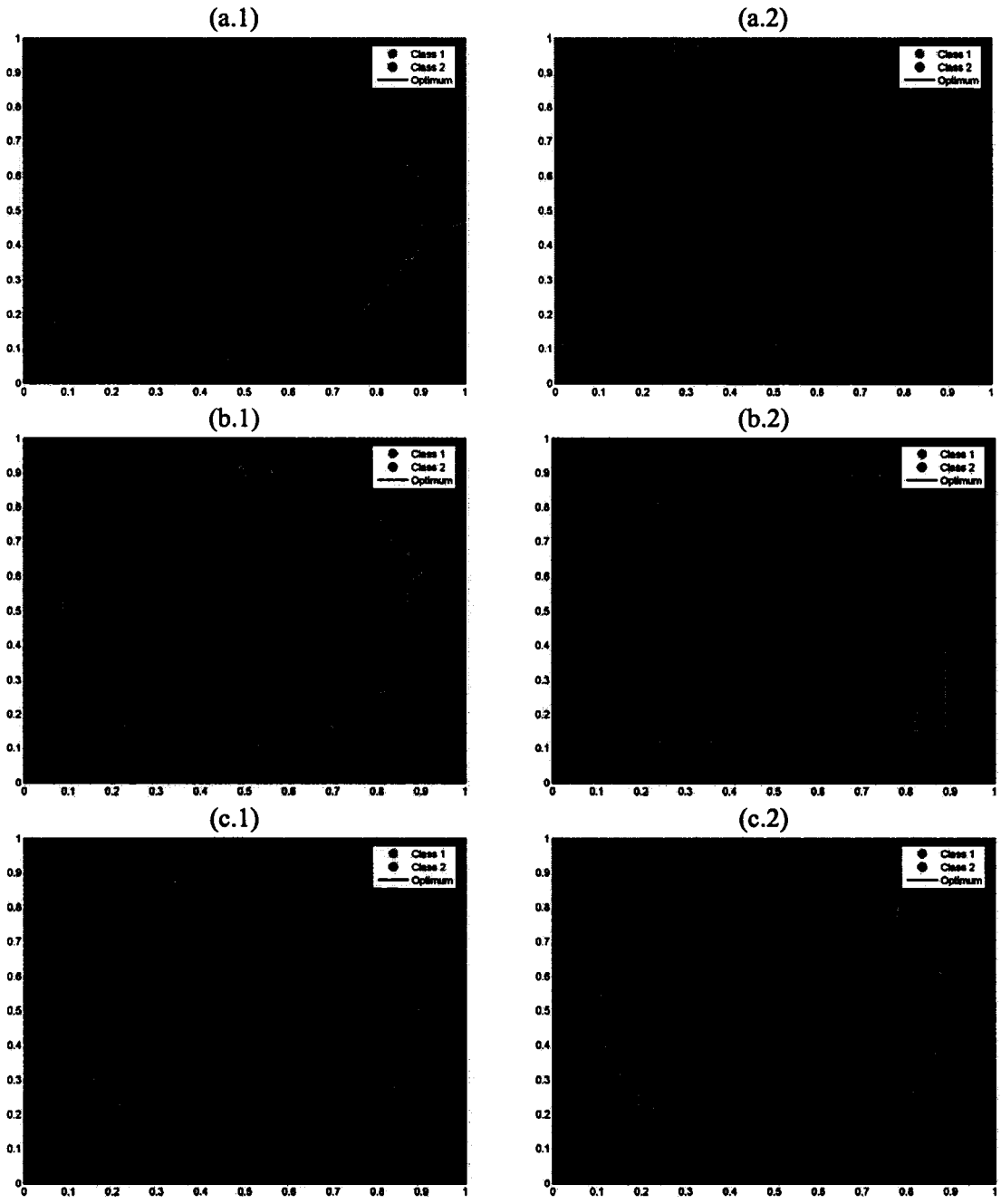


Figure 63 Bornes obtenues lors de l'accroissement de la taille de la base d'apprentissage avec la base DB_{CIS}, avec (PSO(HV)) et sans optimisation (HV MT+)

Soit : (a) 12 patrons, (b) 226 patrons et (c) 5000 patrons d'entraînement par classe, et (.1) avec optimisation des paramètres et (.2) sans optimisation avec MT+.

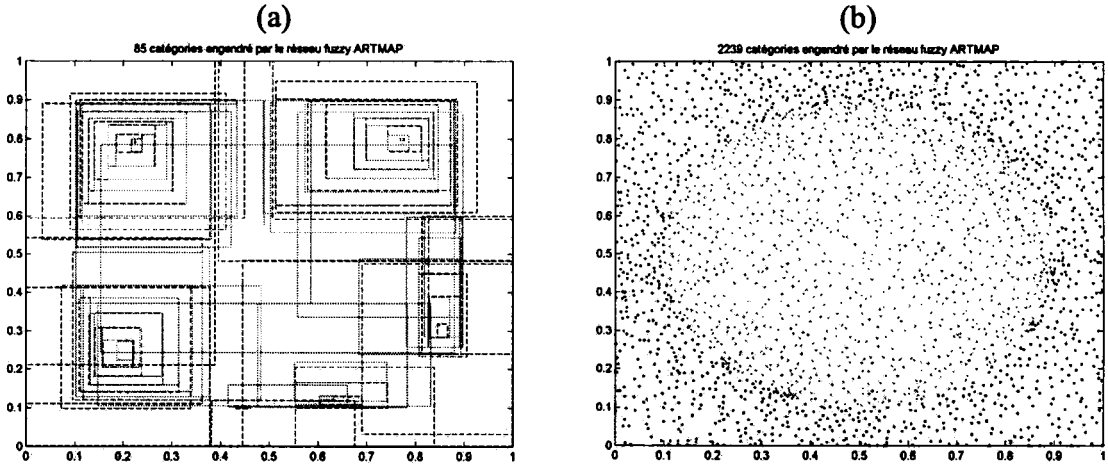


Figure 64 Catégories obtenues avec 5000 p/ω de la base DB_{CIS}
(a) sans optimisation des paramètres HV MT+ et (b) avec optimisation des paramètres PSO(HV)

Ainsi, lors de l'optimisation des paramètres internes des réseaux fuzzy ARTMAP avec PSO(HV) sur la base DB_{CIS}, les paramètres obtenus tendent à créer un très grand nombre de petites catégories, soit 2239 catégories avec l'optimisation (PSO(HV)) comparativement à 85 sans optimisation (HV MT+). On remarque que, sur ces 2239 catégories, seulement 134 représentent plus d'un patron. Ainsi, 94.20% des catégories obtenues avec l'optimisation des paramètres représentent un seul patron. Ce phénomène soulève un fait intéressant, soit que les réseaux fuzzy ARTMAP peuvent tendre vers le comportement d'un classificateur 1NN. Cependant, la fonction de calcul de la distance est différente. En outre, puisque tous les patrons de la base d'entraînement ne deviennent pas des catégories, on peut parler d'un réseau semblable au 1NN mais possédant un meilleur taux de compression. En fait, on peut établir qu'un classificateur 1NN a une compression de 1, soit une catégorie pour chaque patron d'apprentissage. Les réseaux fuzzy ARTMAP obtenant des performances en généralisation semblables à celles de *k*NN *(k est majoritairement égal à 1 avec les bases de données sans chevauchement)* possèdent des compressions allant de 2 à 9 suivant l'augmentation graduelle de la base d'apprentissage avec DB_{CIS} (voir figure 57.c).

4.4 Conclusion

Ce chapitre a clairement montré l'importance d'une sélection adéquate des paramètres internes des réseaux fuzzy ARTMAP en fonction du type de bases de données à classifier. Cependant, puisqu'il n'est pas aisé d'établir des valeurs précises pour chaque cas, les stratégies d'apprentissage spécialisées pour les réseaux FAM que nous avons développées résolvent ce problème. L'algorithme PSO optimise les valeurs des paramètres, obtenant ainsi de meilleures performances en généralisation et ce, pour tous les problèmes que nous avons testés. Il faut également noter que d'autres algorithmes d'optimisation pourraient être utilisés tels que les algorithmes génétiques. D'autre part, les taux de compression obtenus avec les bases possédant un degré de chevauchement sont nettement meilleurs qu'avec l'utilisation des paramètres standard MT- ou MT+. En optimisant ces paramètres, le réseau FAM tend soit vers un estimateur des centres de masses de chaque classe (comme avec les bases synthétiques avec chevauchement), soit vers un réseau semblable au 1NN (comme avec les bases synthétiques sans chevauchement).

La stratégie PSO(1EP) obtient des performances en généralisation et des taux de compression majoritairement similaires aux trois autres stratégies d'apprentissage tout en obtenant les temps de convergence les plus courts et ce, pour toutes les bases de données testées. Il est donc recommandé d'employer uniquement la stratégie PSO(1EP) afin d'accélérer les temps de convergence de l'optimisation.

Finalement, la caractéristique du fuzzy ARTMAP qui engendre le plus de dégradation des performances est la valeur des paramètres. On ne peut donc se fier aux valeurs standard (MT- et MT+) lors de l'utilisation des réseaux fuzzy ARTMAP et l'optimisation des ces paramètres pour ces réseaux devient une nécessité. Cependant, même avec les paramètres optimisés, il reste encore une légère erreur de sur-apprentissage due à la taille de la base d'apprentissage.

Le tableau XI présente un sommaire des résultats obtenus lors de l'utilisation de la taille maximale de la base d'apprentissage avec les différentes stratégies d'entraînement pour les diverses bases de données. Ces résultats sont l'erreur en généralisation moyenne et la dispersion des résultats obtenues sur les 10 réplications. L'annexe 7 présente une synthèse de tous les degrés de chevauchement testés.

Tableau XI

Résultats sommaires avec 5k patrons par classe

Stratégies d'apprentissage	Erreur en généralisation moyenne (dispersion des résultats) %				
	DBμ(1%)	DBμ(9%)	DBμ(25%)	DB_{cls}	DB_{P2}
Erreur théorique	1,00	9,00	25,00	0,00	0,00
CQB	1,00 (0,04)	9,12 (0,08)	25,11 (0,10)	ND	ND
***k*NN**	1,08 (0,03)	9,88 (0,08)	27,23 (0,12)	0,86 (0,03)	1,65 (0,04)
1NN	1,54 (0,03)	13,35 (0,10)	33,49 (0,16)	0,84 (0,02)	1,61 (0,04)
FAM 1EP MT-	2,75 (0,20)	22,49 (0,87)	40,58 (0,47)	4,20 (0,25)	8,89 (0,44)
FAM HV MT-	2,17 (0,08)	20,80 (0,56)	39,83 (0,31)	1,69 (0,07)	4,26 (0,16)
FAM $CONV_W$ MT-	2,74 (0,18)	20,45 (0,46)	40,31 (0,27)	1,77 (0,09)	4,48 (0,38)
FAM $CONV_P$ MT-	2,56 (0,09)	22,00 (0,66)	40,42 (0,31)	1,59 (0,04)	4,51 (0,19)
FAM 1EP MT+	2,51 (0,14)	18,78 (0,38)	38,81 (0,36)	3,98 (0,21)	7,33 (0,33)
FAM HV MT+	1,88 (0,05)	15,17 (0,13)	36,10 (0,20)	1,58 (0,05)	3,68 (0,07)
FAM $CONV_W$ MT+	1,97 (0,09)	15,30 (0,16)	35,94 (0,15)	1,64 (0,05)	3,66 (0,08)
FAM $CONV_P$ MT+	1,90 (0,07)	15,44 (0,15)	36,14 (0,20)	1,47 (0,05)	3,61 (0,08)
FAM PSO(1EP)	1,04 (0,03)	9,35 (0,08)	25,50 (0,09)	1,13 (0,12)	2,05 (0,10)
FAM PSO(HV)[3]	1,07 (0,04)	9,24 (0,08)	25,27 (0,09)	1,06 (0,05)	1,99 (0,04)
FAM PSO($CONV_W$)[3]	1,06 (0,03)	9,32 (0,08)	25,56 (0,16)	1,02 (0,05)	2,02 (0,04)
FAM PSO($CONV_P$)[3]	1,06 (0,04)	9,31 (0,07)	25,48 (0,18)	1,00 (0,03)	2,01 (0,03)

[3] Résultats obtenus avec 3940 patrons par classe au lieu de 5000 patrons par classe.

CHAPITRE 5

RÉSULTATS AVEC LA BASE RÉELLE NIST SD19

Ce chapitre présente les résultats obtenus lors de l'utilisation de la base de données réelles NIST SD19 [29].

Plusieurs aspects sont traités dans ce chapitre. L'effet de la technique de normalisation est tout d'abord présenté. Cette section permet d'observer les différences entre les deux techniques de normalisation que nous utilisons avec la base de données NIST SD19, et ainsi choisir laquelle sera employée pour toutes les autres simulations. La seconde section présente les résultats obtenus avec les stratégies d'apprentissage normalement utilisées dans la littérature. Ces résultats permettent de quantifier les performances obtenues lors de l'utilisation des stratégies spécialisées optimisant les paramètres internes FAM. En troisième partie, les effets engendrés par la polarité du MatchTracking sont abordés. La quatrième section présente les résultats obtenus lors de l'utilisation des stratégies d'apprentissage spécialisées pour les réseaux fuzzy ARTMAP. Finalement, une conclusion termine ce chapitre.

5.1 Effets de la technique de normalisation

Lors de l'utilisation de la base NIST SD19, nos vecteurs de données sont constitués de 132 caractéristiques, toutes comprises entre 0 et 1 inclusivement. Tel qu'énoncé dans la section 2.4.1, il est possible d'effectuer une normalisation des données même si celles-ci sont déjà comprises dans l'intervalle [0, 1], et ce, afin d'améliorer leur répartition sur cet intervalle. Cette section présente les effets de la technique de normalisation sur la base de données réelles. Bien qu'avec les données synthétiques aucune différence n'ait été observée entre les deux techniques de normalisation, il est intéressant de voir si cela est toujours le cas avec les caractéristiques extraites de la base de données réelles NIST SD19.

La Figure 65 présente l'histogramme des valeurs de la 15^{e} caractéristique de la base NIST SD19, sans normalisation, avec les 150 000 patrons de la base d'apprentissage. Tel qu'on peut le constater, la majorité des valeurs est située près de 0. Cette caractéristique représente bien la distribution des 132 caractéristiques comprises dans la base NIST SD19 car la grande majorité d'entre elles ont un histogramme similaire à la caractéristique #15.

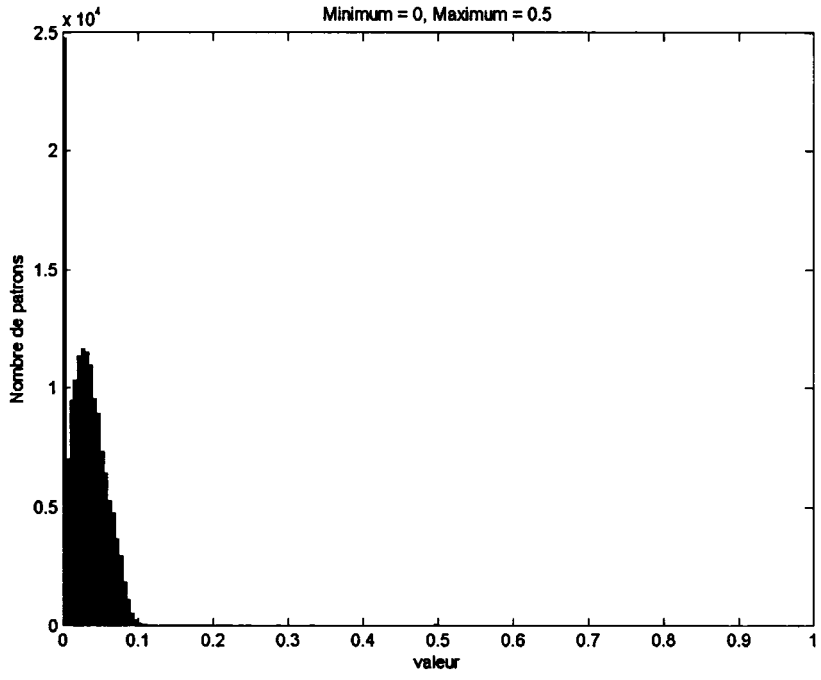


Figure 65 NIST SD19, histogramme de la caractéristique #15 sans normalisation

Lors de l'application de la technique de normalisation Centrée Réduite (voir section 2.4), il faut appliquer l'équation (2.7) pour chaque caractéristique. La dimension #15 a une moyenne de 0,032280 et une variance de 0,000644. Lors de l'utilisation de données provenant d'une distribution normale, pour obtenir 99% des patrons dans l'intervalle [-1, 1], le dénominateur de l'équation (2.7) doit être remplacé par $3 \cdot \sigma_i$. La Figure 66 présente l'histogramme de la 15^{e} caractéristique après avoir effectué la normalisation avec l'équation (2.7) avec le dominateur $3 \cdot \sigma_i$.

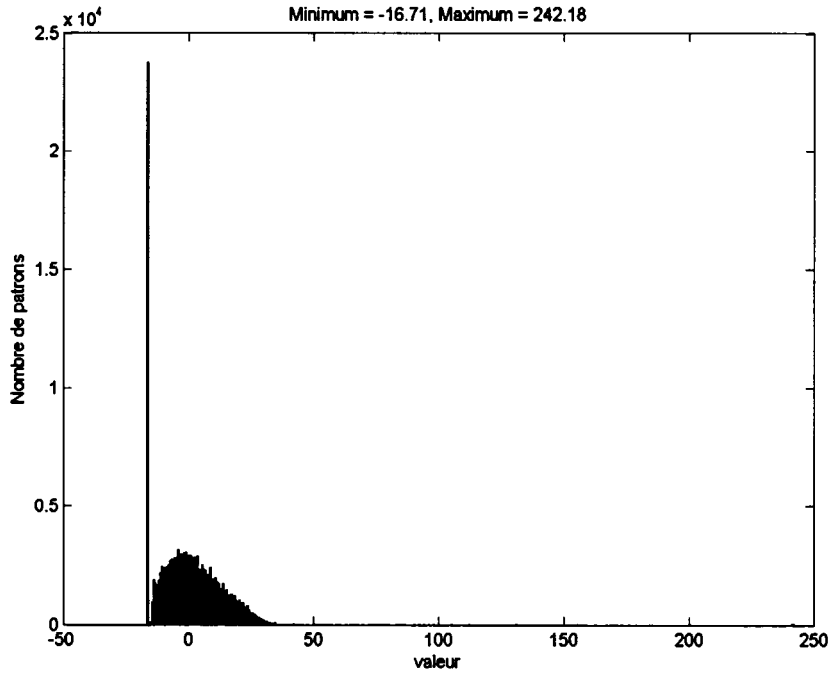


Figure 66 NIST SD19, #15 - application de l'équation (2.7)

Un problème survient lors de la division par la variance. Celle-ci étant très petite, les patrons ayant des valeurs légèrement différentes de la moyenne se font octroyer une forte valeur normalisée, laquelle est rapidement projetée en dehors de l'intervalle [-1, 1].

Sur les 150 000 patrons constituant la base d'entraînement, 65 734 patrons se retrouvent au dessus de la borne +1 et 75 442 patrons se retrouvent en dessous de la borne -1. Ainsi, 141 176 patrons sont en dehors de l'intervalle [-1, 1] et sont mis à la borne la plus proche. La Figure 67 présente l'histogramme final obtenu lors de l'application complète de la technique de normalisation Centrée Réduite sur la 15^{e} caractéristique de la base de données NIST SD19.

Malheureusement, ce phénomène n'est pas unique à la 15^{e} caractéristique. En moyenne, 8 525 patrons sont compris dans l'intervalle]0 1[pour chacune des 132 caractéristiques.

Ainsi, 94.32% des données normalisées avec la technique Centré Réduite sont à l'une des deux bornes, soit 0 ou 1.

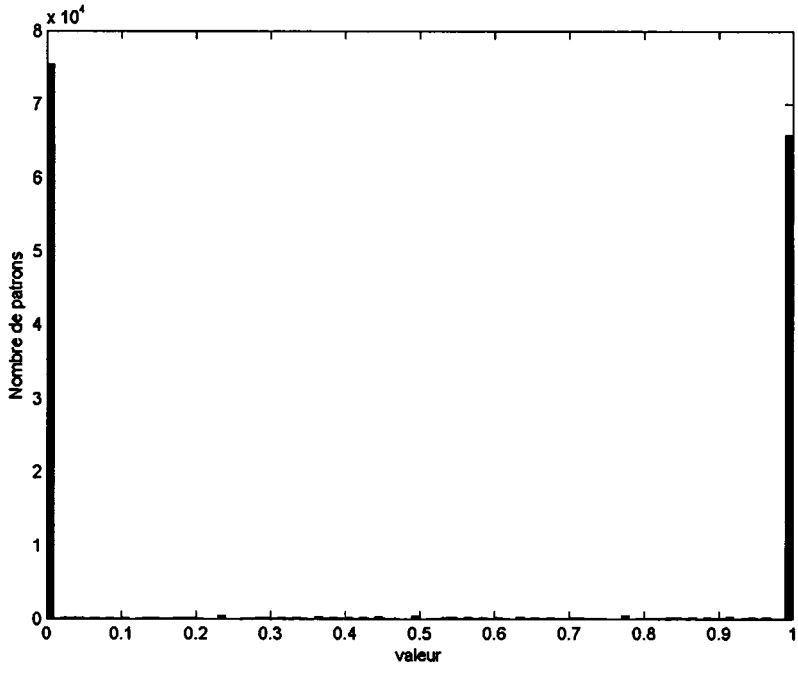


Figure 67 NIST SD19, #15 - Normalisation Centrée Réduite

Regardons maintenant l'effet de l'application de la technique de normalisation MinMax sur les caractéristiques extraites de la base NIST SD19, toujours par rapport à la 15^{e} caractéristique.

Pour appliquer la méthode de normalisation MinMax (voir l'équation (2.6)), il faut connaître la valeur minimale et maximale de chaque dimension. La valeur minimale de la 15^{e} caractéristique est de 0 et sa valeur maximale de 0,500. Lors de l'utilisation de (2.6), les données normalisées sont automatiquement comprises dans l'intervalle [0, 1]. La figure 68 présente la normalisation MinMax effectuée sur la 15^{e} caractéristique de la base de données NIST SD19. On peut constater que l'aspect général de la répartition des données non normalisées (voir figure 65) est conservé lors de cette normalisation, mais

que les données s'étendent sur toute la plage [0, 1]. La zone où les données sont concentrées est passée de [0, 0.1] (original, voir figure 65) à [0, 0.2] (voir figure 68) lors de l'application de la normalisation MinMax.

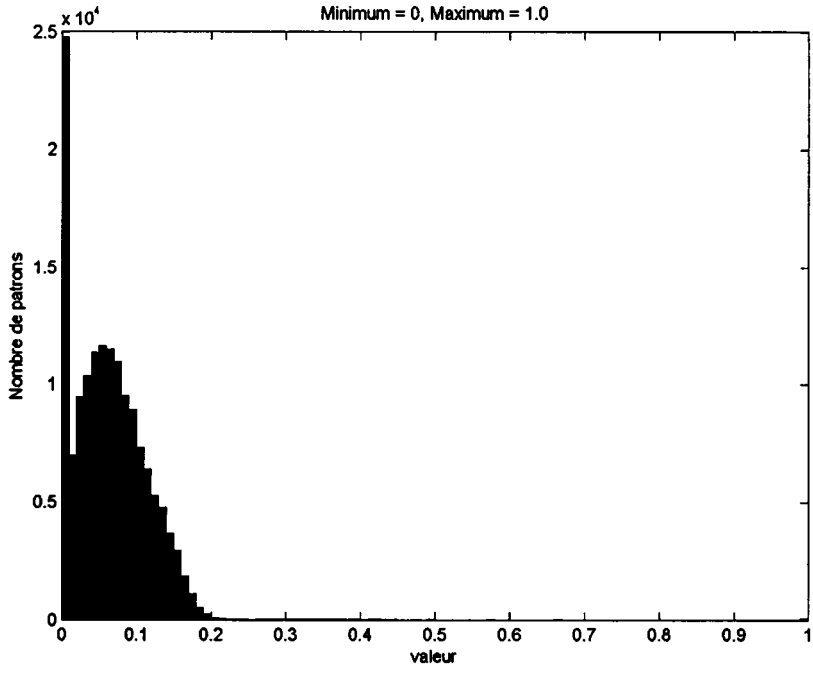


Figure 68 Normalisation MinMax de la 15^{e} caractéristique de la base NIST SD19

Nous ne pouvons choisir une technique de normalisation sur la base NIST SD19 uniquement à l'aide de ces figures. Cependant, plusieurs points laissent prévoir une moins bonne performance avec la méthode de normalisation Centrée Réduite. Avec la normalisation MinMax, l'aspect général des données non normalisées est conservé, alors qu'avec la normalisation Centrée Réduite, il ne l'est pas. Ainsi, la normalisation Centrée Réduite sur la base de données NIST SD19 peut causer une perte d'information contenue dans la variabilité des données.

L'importance de cette perte d'information sera représentative des performances obtenues en généralisation sur les réseaux FAM avec la normalisation Centrée Réduite.

5.1.1 Résultats

Cette section présente les résultats obtenus lors du test entre les deux techniques de normalisation. Les résultats de cette section permettent de sélectionner une seule technique de normalisation pour l'ensemble des simulations restantes concernant la base de données NIST SD19.

La Figure 69 présente les résultats obtenus pour les deux techniques de normalisation, lors de l'augmentation graduelle de la taille de la base d'apprentissage (voir tableau VIII, section 2.1.2), avec la stratégie d'apprentissage HV sur la base de test hsf_7. Les erreurs en généralisation, les taux de compression ainsi que les temps de convergence y sont présentés. Les résultats obtenus sans normalisation (données brutes) ainsi que les performances du kNN (k = 1) sont également présentés pour fins de comparaison.

5.1.2 Analyse

Même si les deux techniques de normalisation obtiennent des performances en généralisation similaires lors de l'utilisation complète de la base d'apprentissage, la normalisation MinMax semble mieux indiquée pour la base de données NIST SD19. L'utilisation de la normalisation MinMax produit de meilleures performances en généralisation lors de l'utilisation des petites et moyennes tailles de la base d'apprentissage. De plus, elle obtient des taux de compression supérieurs à ceux de la normalisation Centrée Réduite. Plus la taille de la base d'apprentissage grandit, plus cet effet est visible. Et finalement, les temps de convergence sont plus petits avec la normalisation MinMax qu'avec la normalisation Centrée Réduite. Pour toutes ces raisons, nous avons sélectionné la normalisation MinMax pour effectuer l'ensemble des simulations restantes concernant la base de données NSIT SD19.

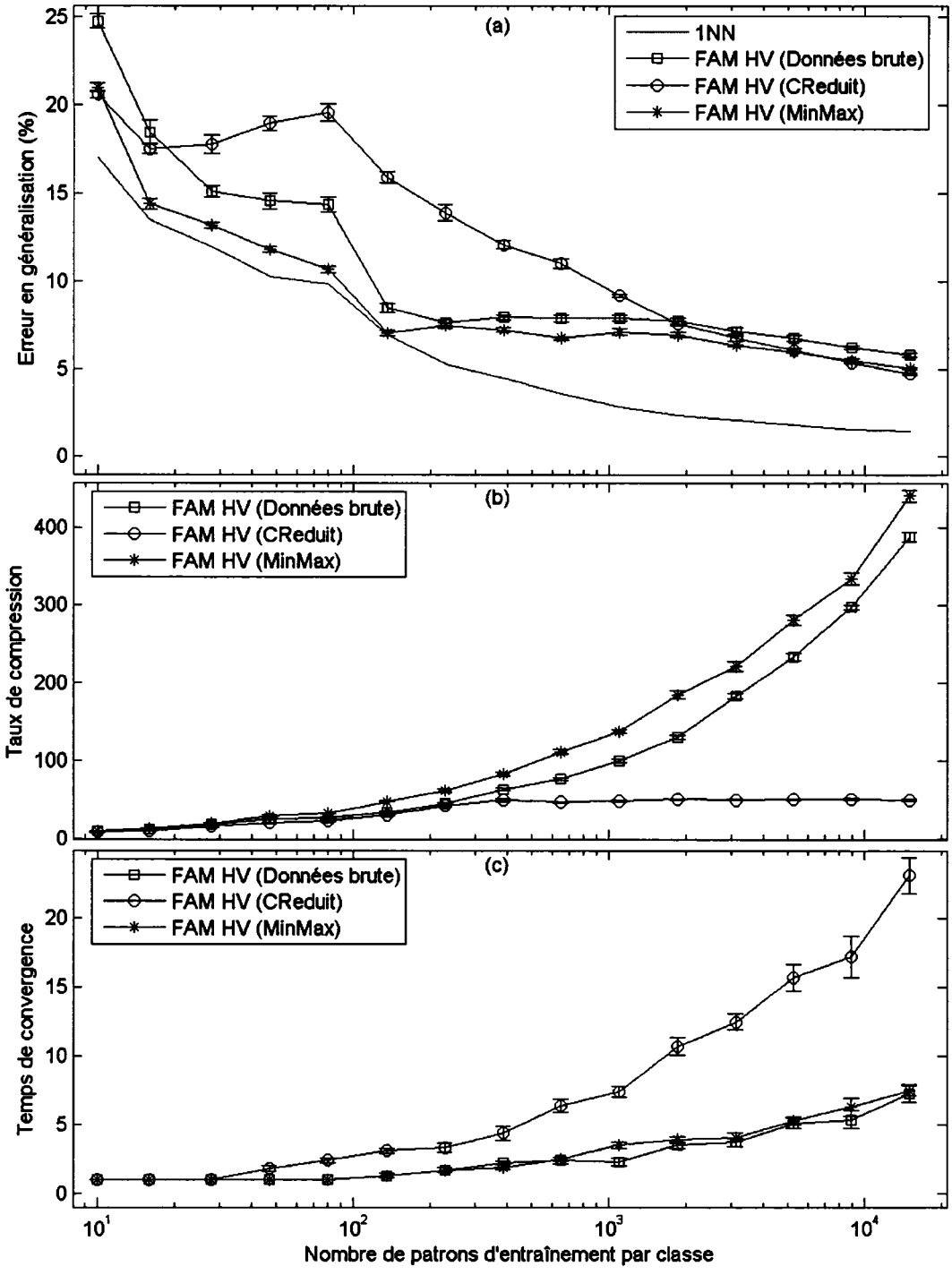


Figure 69 Comparaison des techniques de normalisation pour NIST SD19, hsf_7

(a) Erreur en généralisation sur hsf_7, (b) temps de convergence et (c) taux de compression

5.2 Stratégies d'apprentissage avec les paramètres standard

Cette section présente les résultats obtenus lors des simulations de la base NIST SD19 avec les quatre stratégies d'apprentissage standard présentées à la section 1.2 et l'augmentation graduelle de la taille de la base d'entraînement. Ces simulations sont effectuées avec les paramètres standard MT- des réseaux fuzzy ARTMAP.

5.2.1 Résultats

Les réseaux fuzzy ARTMAP obtenus avec la base NIST SD19 sont testés par deux bases de test, soit hsf_7 et hsf_4. La base de test hsf_4 représente une base de test bruitée et donc les performances en généralisation sur cette base seront moins bonnes que sur la base hsf_7, qui elle ne contient aucune distorsion. La Figure 70 présente les erreurs en généralisation évaluées sur hsf_7 ainsi que les taux de compression et les temps de convergence obtenus.

5.2.2 Analyse

Tel qu'on peut le voir à la Figure 70, les erreurs en généralisation ont la même tendance que lors des simulations avec les bases de données synthétiques sans chevauchement, soit de diminuer avec l'augmentation du nombre de patrons d'apprentissage. Par contre, au-delà d'une certaine taille de la base d'entraînement, soit environ 136 patrons par classe (1360 au total), les erreurs en généralisation obtenues par les réseaux fuzzy ARTMAP diminuent de plus en plus lentement. Cependant, la compression continue de croître linéairement tout au long de l'augmentation de la taille de la base d'entraînement. On remarque également que, comme nous l'avons observé dans le cas des bases de données synthétiques sans chevauchement, les réseaux *k*NN obtiennent de meilleures performances avec $k = 1$. De plus, les temps de convergence pour la stratégie d'apprentissage $CONV_P$ obtiennent souvent le nombre maximal d'époques (1000 époques). Il s'agit du même phénomène que nous avons observé à la section 3.4.3lequel est dû à l'utilisation de MT-.

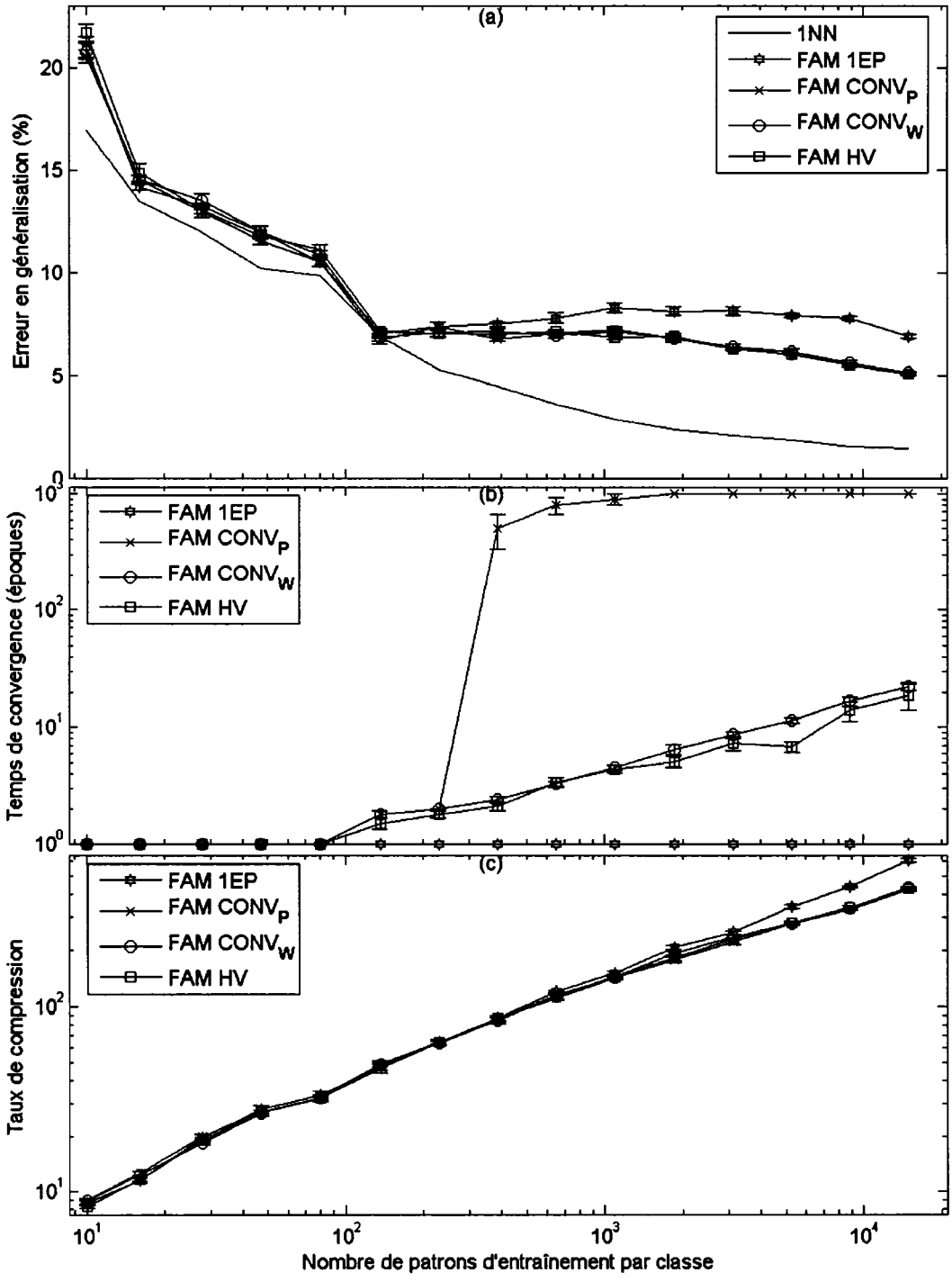


Figure 70 Performances du FAM (MT-) en fonction de la taille de la base d'apprentissage avec la base NIST SD19
(a) Erreur en généralisation sur hsf_7, (b) temps de convergence et (c) taux de compression

À partir de ces résultats, aucun effet de sur-apprentissage généré par le nombre d'époques d'entraînement n'est visible. Au niveau de la dégradation des performances en fonction de la taille de la base d'apprentissage, il y a un effet de stagnation des performances en généralisation entre 229 à 1856 patrons d'entraînement par classe. Cependant, après 1856 patrons, l'erreur en généralisation recommence à diminuer. Ce palier indique une dégradation de performance causée par la taille de la base d'apprentissage sur ce palier. Ce palier est plus visible avec la stratégie d'entraînement 1EP. Par contre, aucune dégradation n'est observée pour les taux de compression.

Les taux de compression augmentent linéairement avec le nombre de patrons d'entraînement générant des taux de compression moyens de 425.77 (±5.25) avec la taille maximale de la base d'apprentissage. Ceci indique que le nombre de catégories augmente, lui aussi, linéairement avec le nombre de patrons d'apprentissage atteignant 352.3 (±15.77) catégories avec l5000 patrons d'entraînement par classe.

Bref, le comportement du FAM semble confirmer que les caractéristiques extraites de la base NIST SD19 possèdent peu ou aucun degré de chevauchement. Ainsi, comme avec les bases DB_{P2} et DB_{CiS}, plus il y aura de patrons dans la base d'entraînement, meilleurs seront les résultats.

5.3 Effets de la polarité du MatchTracking

Cette section présente l'influence de la polarité du MatchTracking. Nous pourrons ainsi voir comment les performances des réseaux fuzzy ARTMAP sont influencées par la polarité de ce paramètre, et ainsi démontrer que les performances des réseaux FAM peuvent être améliorées ou diminuées selon le type de MT.

5.3.1 Résultats

La Figure 71 présente la comparaison des résultats entre MT- et MT+ pour la base de données NIST SD19 au niveau des erreurs en généralisation, des taux de compression ainsi que des temps de convergence. Lorsque la courbe est positive, la valeur obtenue avec MT- est plus grande que celle obtenue avec la base MT+, et vice-versa.

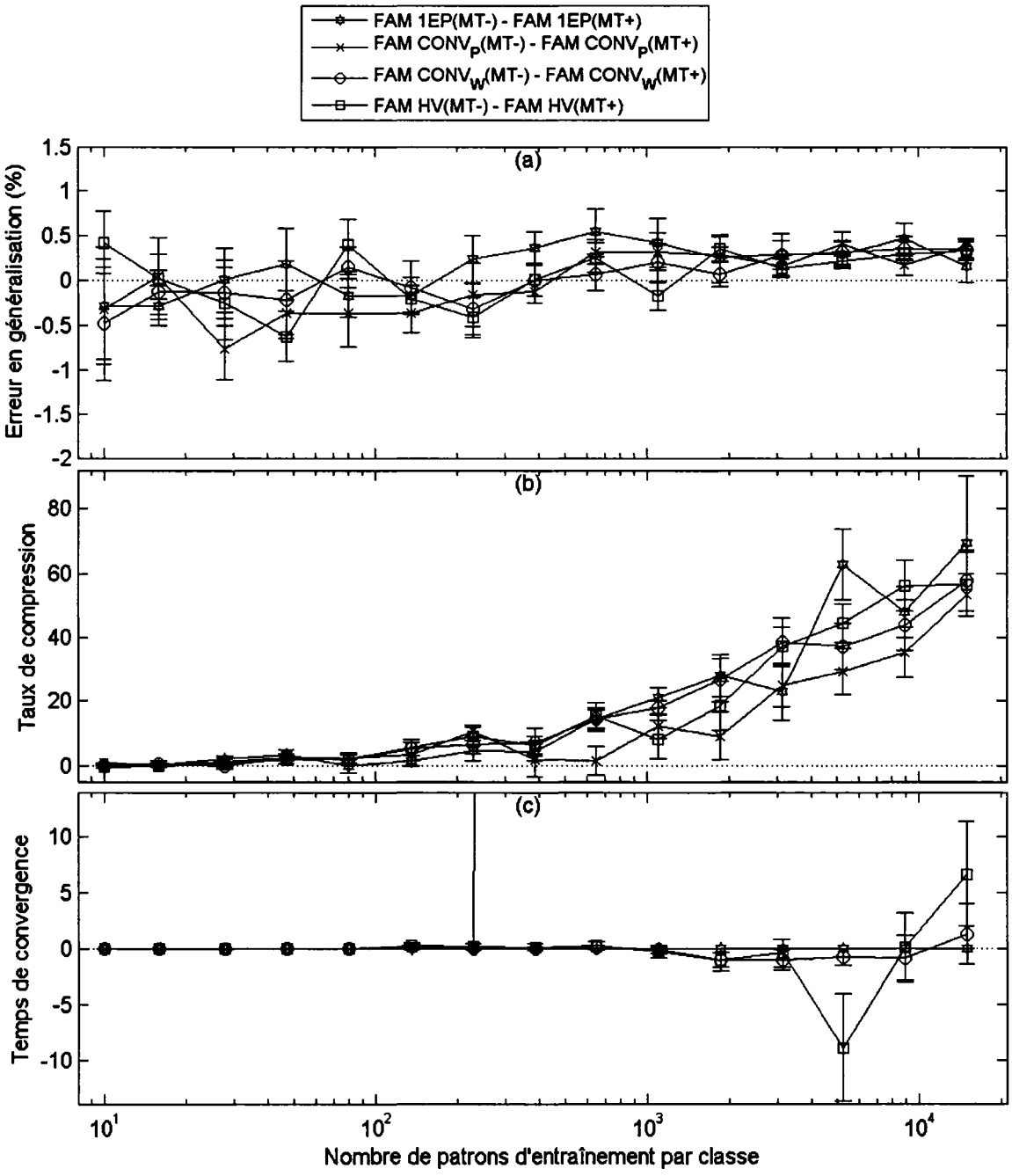


Figure 71 Différence entre MT- et MT+ avec la base NIST SD19

(a) Erreur en généralisation, (b) Taux de compression, (c) Temps de convergence

5.3.2 Analyse

En analysant les résultats présentés à la Figure 71, nous remarquons que les erreurs en généralisation ont tendance à être légèrement supérieures avec MT- lors de l'utilisation des grandes tailles de la base d'apprentissage. Par contre, MT- obtient des taux de compression supérieurs à ceux obtenus avec MT+. Plus la taille de la base d'entraînement augmente, plus cet écart s'accroît. Au niveau des temps de convergence, les stratégies 1EP, $CONV_W$ et HV obtiennent des résultats similaires avec les deux types de MatchTracking.

Lors de l'utilisation de la stratégie $CONV_P$ les différences obtenues pour le temps de convergence sont très grandes, soit proche de 1000 époques. Cette courbe est présentée à la Figure 72. Cette grande différence correspond au même phénomène observé à la section 3.4.3.

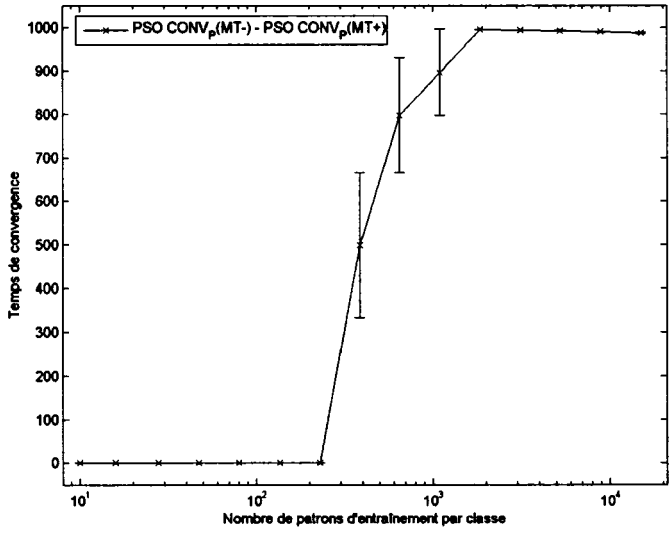


Figure 72 Différence entre $CONV_P$(MT-) et $CONV_P$(MT+) avec NIST SD19

Ainsi, tout comme avec les bases de données synthétiques sans chevauchement (DB_{P2} et DB_{CIS}), il est difficile de sélectionner un type de MatchTracking au niveau des

performances en généralisation. Par contre, puisque MT- obtient de meilleurs taux de compression, MT- semble mieux indiqué pour la base NIST SD19. En utilisant l'optimisation des paramètres, nous pourrons voir quelle polarité du MatchTraking sera utilisée par les réseaux obtenus avec les stratégies d'apprentissage spécialisées.

5.4 Stratégies d'apprentissage spécialisées

Cette section présente les résultats obtenus avec la base de données NIST SD19 lors de l'optimisation des paramètres internes des réseaux fuzzy ARTMAP avec la stratégie d'apprentissage spécialisée PSO(1EP). Puisque le chapitre 4 a conclu que nous pouvions diminuer le temps de convergence lors de l'optimisation des paramètres en utilisant la stratégie PSO(1EP) sans pour autant diminuer les performances obtenues, nous utiliserons cette stratégie avec la base de données NIST.

Lorsque nous avons appliqué la stratégie d'apprentissage spécialisée PSO(1EP), avec les plage de valeurs des paramètres : $\alpha \in [0.00001, 1]$, $\bar{\rho} \in [0, 1]$, $\beta \in [0, 1]$ et $\varepsilon \in [-1, 1]$, nous n'avons pu effectuer toutes les simulations prévues par le protocole expérimental et la plage d'optimisation du paramètre $\bar{\rho}$ a été réduite à [0 0.90]. La section suivante présente les résultats ainsi que leurs analyses, tout en indiquant la raison pour laquelle nous n'avons pu compléter les simulations avec la plage complète du paramètre $\bar{\rho}$.

5.4.1 Résultats

Cette section présente les résultats obtenus lors de l'optimisation des paramètres des réseaux fuzzy ARTMAP avec $\bar{\rho} \in [0, 1]$ et $\bar{\rho} \in [0, 0.90]$, lors l'utilisation de la stratégie d'apprentissage PSO(1EP), et ce pour les erreurs en généralisation et les taux de compression. Les résultats obtenus avec kNN ($k = 1$) et avec la stratégie d'entraînement typique HV (MT-) sont également présentées pour fins de référence.

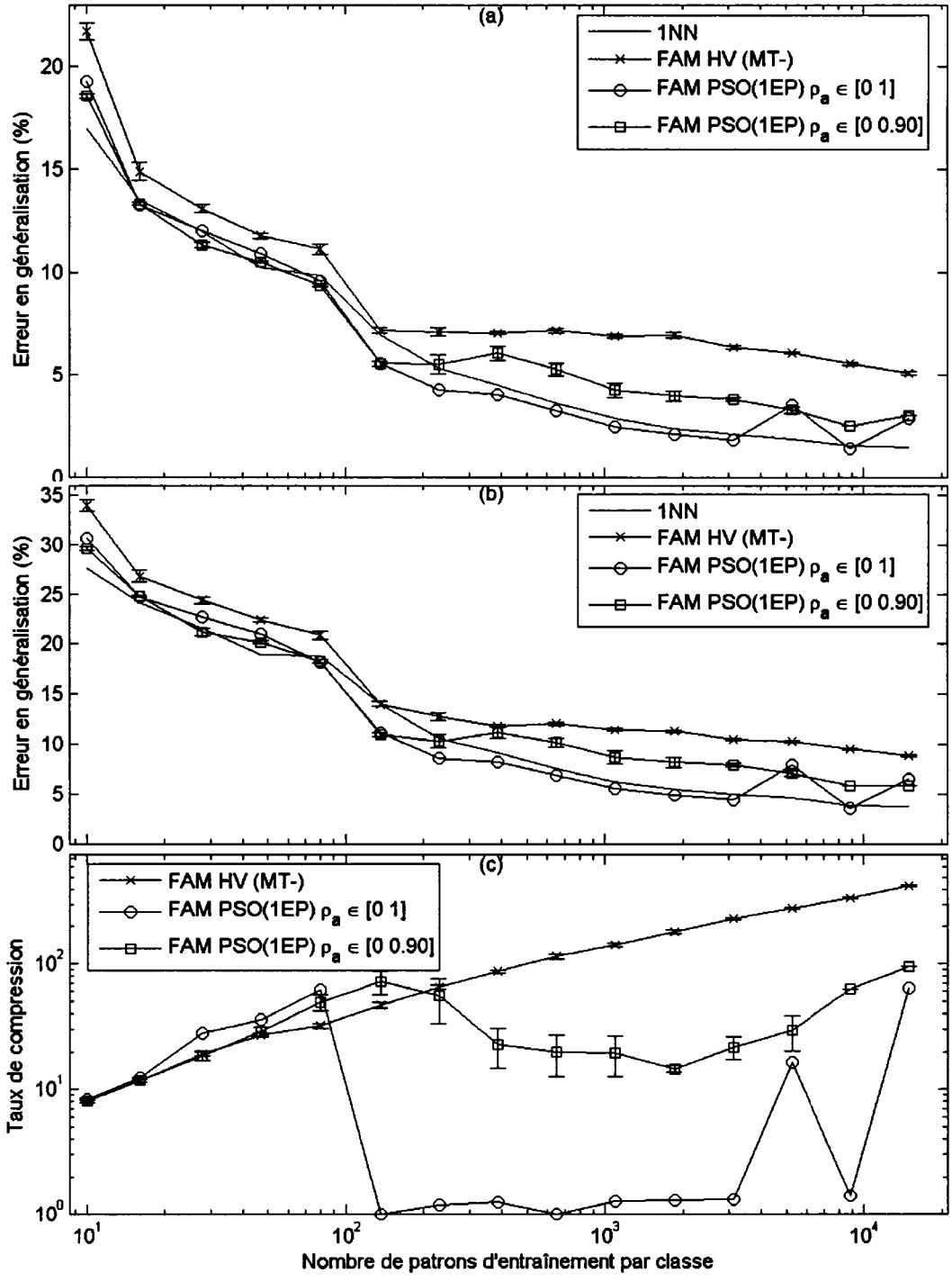


Figure 73 Performance du FAM lors de l'optimisation des paramètres avec PSO(1EP) pour la base NIST SD19
(a) Erreur en généralisation sur hsf_7, (b) Erreur en généralisation sur hsf_4, et (c) taux de compression

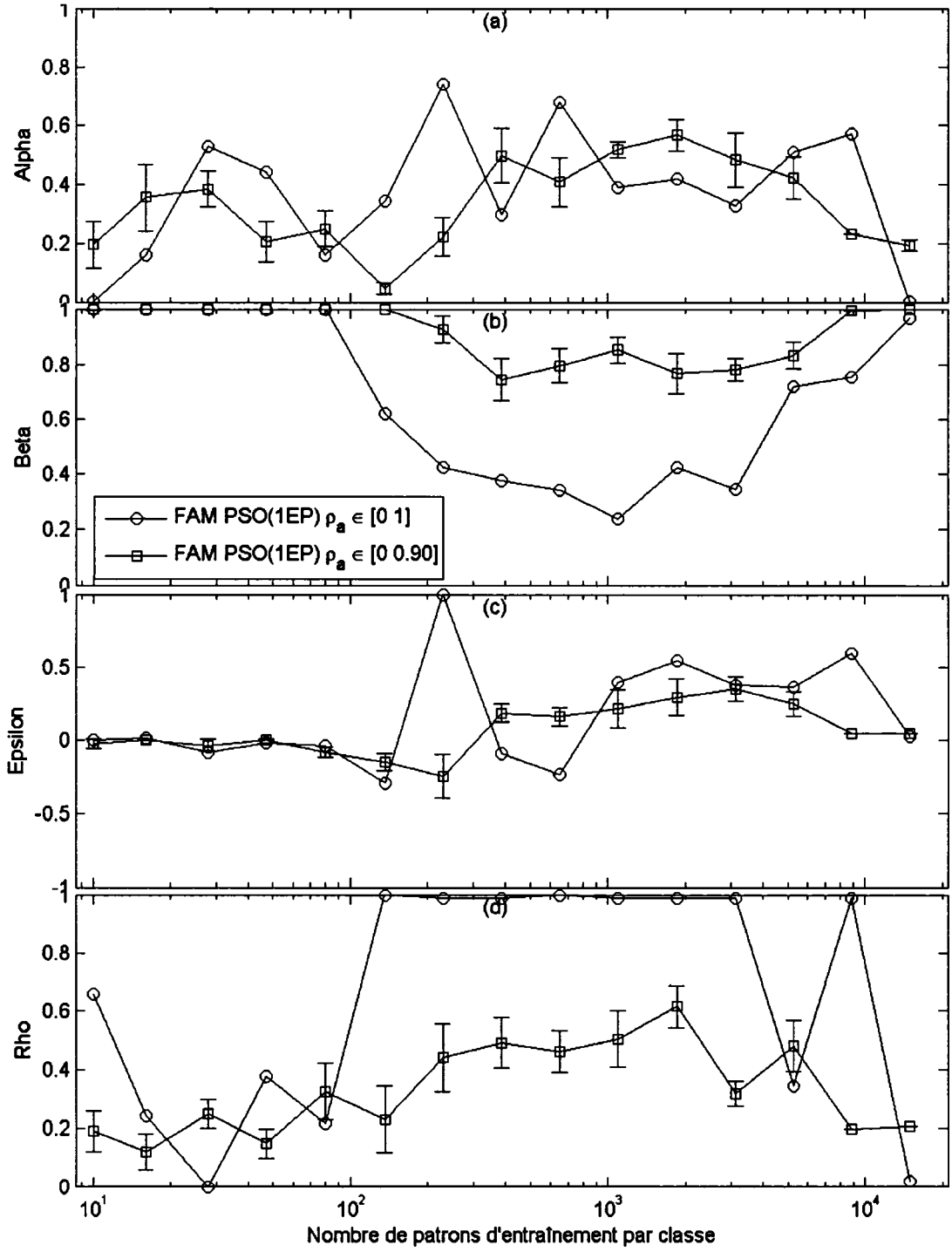


Figure 74 Valeurs des paramètres internes du FAM lors de l'utilisation de la stratégie spécialisée PSO(1EP) avec la base NIST SD19

(a) α: paramètre de choix, (b) β : vitesse d'apprentissage, (c) ε : paramètre de MatchTracking et (d) $\overline{\rho}$: vigilance de base

5.4.2 Analyse

Tel que nous pouvons le constater, les résultats obtenus lors de l'optimisation des paramètres sont très proches de ceux observés avec le 1NN. Les performances en généralisation sont meilleures que celles obtenues avec les stratégies HV MT- et HV MT+. Les résultats avec la plage $\overline{\rho} \in [0, 1]$ sont généralement meilleurs, avec les petites tailles de base d'apprentissage, que ceux de *k*NN. L'écart entre leurs performances diminue lorsque la taille d'apprentissage augmente, pour finalement disparaître. Deux résultats de cette courbe obtiennent des erreurs en généralisation supérieures au 1NN, soit avec 52 860 et 150 000 patrons par classe. Ceci est dû au fait que nous avons réussi à exécuter seulement deux des quatre itérations PSO avec la base NIST SD19, et ce pour une seule des 10 répétitions. En effectuant les quatre itérations PSO ces résultats rejoindront ceux obtenus avec 1NN.

Cependant, il faut noter que les erreurs en généralisation obtenues avec les réseaux FAM sont supérieures à celles obtenues avec les réseaux MLP (Multi-Layer Perceptron) et SVM (Support Vector Machine) [38]. Les avantages du réseau FAM sont au niveau des performances en généralisation lors de l'utilisation de bases ne possédant que peu de données, de la possibilité d'exécuter un apprentissage incrémental, ainsi que de la rapidité de la classification.

Lors de l'optimisation avec $\overline{\rho} \in [0, 1]$, les temps de simulation sont très grands et nous n'avons pu continuer ces simulations car nous manquions de puissance de calcul. Ces temps de calcul importants sont provoqués par le très grand nombre de catégories créées avec la base NIST SD19. Pour bien illustrer ce problème, nous avons étudié l'effet des paramètres $\overline{\rho}$ et ε lors de l'utilisation de 3870 patrons d'entraînement avec la base NIST SD19. La Figure 75 montre le ratio temporel de l'entraînement avec la stratégie 1EP pour 3870 patrons d'apprentissage avec la base NIST SD19 et ce, avec les paramètres $\alpha = 0.01$, $\beta = 1.0$, $\varepsilon \in [-1, 1]$ et $\overline{\rho} \in [0, 1]$.

Tel que montré à la Figure 75, en changeant la valeur des paramètres $\bar{\rho}$ et ε, un réseau peut prendre jusqu'à 800 fois le temps d'apprentissage minimum obtenu. Lors de l'utilisation des paramètres standard MT-, nous obtenons des temps presque égaux au temps d'apprentissage minimum. Avec la stratégie d'apprentissage 1EP MT-, nous obtenons un temps d'apprentissage de 14.11 secondes pour les 3870 patrons d'entraînement de la base NIST SD19. *(À noter que ces simulations ont été effectuées sur un Atlhon AMD 2800+).* Si la valeur du paramètre $\bar{\rho}$ est plus grande que 0.90, le temps d'entraînement du réseau, toujours avec la stratégie d'apprentissage 1EP MT-, peut prendre jusqu'à 3 heures et 8 minutes.

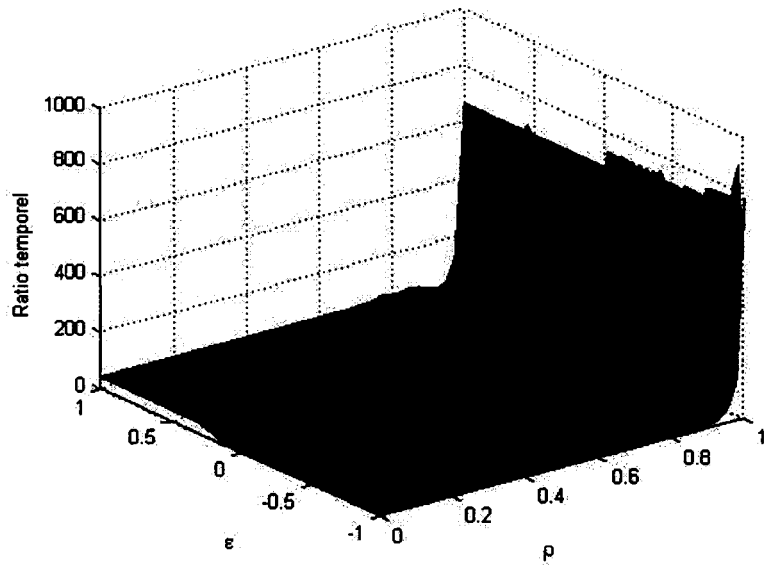


Figure 75 Ratio temporel du temps d'entraînement avec 1EP sur NIST SD19 pour 3870 patrons d'apprentissage

Le nombre minimum de réseaux FAM entraînés pour effectuer une optimisation PSO est de 660 (4 répétitions, 15 particules PSO et 11 itérations PSO). Cette optimisation est répétée 10 fois pour obtenir des résultats moyens, ce qui nous donne un minimum de 6600 réseaux pour chaque taille de la base d'entraînement. Sur ces 6600 réseaux, si

seulement 5% d'entre eux obtiennent des valeurs de paramètres FAM avec un $\overline{\rho} > 0.90$, le temps d'apprentissage requis pour cette optimisation sera environ de 44.1 jours. Cependant, l'optimisation PSO requiert généralement plus d'itérations de recherche que le nombre minimum, augmentant ainsi le nombre de réseaux FAM à entraîner. De plus, si l'optimisation trouve un optimum dans la région critique ($\overline{\rho} > 0.90$), il y aura bien plus de 5% des réseaux utilisés par l'optimisation dans cette région. Bref, le temps requis pour l'optimisation de la base NIST SD19 demande une très grande puissance de calcul. Il est donc très avantageux d'utiliser une grappe d'ordinateurs pour calculer ces optimisations.

Cependant, malgré l'utilisation de cluster d'ordinateurs, le temps d'entraînement requis lors de l'utilisation des 150000 patrons d'apprentissage avec la stratégie 1EP MT- est de 2 minutes et 1 secondes. Ceci se traduit par un réseau pouvant prendre jusqu'à 26.85 heures de temps de convergence si $\overline{\rho} > 0.90$. Ce qui entraîne un temps minimal d'environ 24.5 jours pour effectuer une des 10 répétitions sur un cluster composé de 15 nœuds.

Lors de l'utilisation de bases de petites tailles, les résultats avec les deux plages d'optimisation du paramètre $\overline{\rho}$ obtiennent des performances en généralisation semblables. Cependant, à partir d'une certaine taille, soit environ 3870 patrons par classe, les résultats obtenus avec la plage complète de $\overline{\rho}$, soit $\overline{\rho} \in [0, 1]$, obtiennent de meilleures performances que lors de la réduction de la plage d'optimisation. Le Tableau XII présente les valeurs des paramètres obtenues lors de l'optimisation des paramètres internes des réseaux fuzzy ARTMAP sur la base de données réelles avec 3870 patrons d'entraînement pour dix répétitions.

Tel qu'on peut le voir, 5 des 10 réseaux optimisés possèdent une valeur de $\overline{\rho}$ proche de 1. En tenant compte de la Figure 75 et du Tableau XII, nous avons décidé que, pour

réduire le temps de calcul, nous devions réduire la plage de recherche pour le paramètre $\overline{\rho}$. Ainsi, la plage de recherche de ce paramètre est limitée à [0, 0.90]. Ceci nous permettra d'obtenir des résultats dans des délais raisonnables.

Cependant, si nous avions une très grande ressource de calcul à notre disposition, nous n'aurions pas eu besoin de faire cette réduction. Tel que le montre la figure 73, la réduction de la plage du paramètre $\overline{\rho}$ entraîne une perte de performance. Cette perte de performance est de plus en plus apparente lors de l'augmentation du nombre de patrons d'apprentissage. Ceci provient du fait que, plus la taille de la base d'entraînement augmente, plus la valeur optimum du paramètre $\overline{\rho}$ se rapproche de 1.

Tableau XII

Valeur des paramètres optimisés avec 3870 patrons d'entraînement sur NIST SD19

OPTIMISATION	α	β	ε	$\overline{\rho}$
1	0,409456	0,816064	0,560486	0,199149
2	0,55138	1	-0,890152	0,922744
3	0,291144	0,776226	0,58208	0,296544
4	0,847107	1	0,474937	0,938798
5	0,132677	0,478892	0,656631	0,99
6	0,469334	0,40215	0,02185	0,99
7	0,190753	0,433142	0,199164	0,99
8	0,321796	0,480391	0,265557	0,99
9	0,681429	0,358223	-0,202317	0,99
10	0,505303	0,399923	0,467291	0,481631

On remarque qu'avec les petites tailles de la base d'entraînement, le fait de réduire la plage des valeurs du paramètre $\overline{\rho}$ n'influence pas les performances obtenues avec

l'optimisation des paramètres. Par contre, lorsque la taille de la base s'approche de 387 patrons d'entraînement par classe, les performances en généralisation sont meilleures avec la plage complète de $\overline{\rho}$ qu'avec la plage réduite. Cet effet est présent avec les deux bases de tests.

La Figure 74 présente l'évolution des valeurs des paramètres internes des réseaux FAM lors de leur optimisation. Les paramètres α et β suivent sensiblement les mêmes tendances avec et sans la plage complète de $\overline{\rho}$. Cependant, au-delà de 229 patrons par classe, les valeurs du paramètre ε ont tendance à diminuer plus vite et $\overline{\rho}$ a tendance à augmenter plus rapidement que lors de l'utilisation de la plage réduite de $\overline{\rho}$. Lors de l'utilisation de la plage [0 0,90] pour le paramètre $\overline{\rho}$, ε tend à rester très élevé pour compenser les valeurs plus faibles de $\overline{\rho}$, alors qu'avec la plage complète de $\overline{\rho}$, ε diminue pour laisser augmenter $\overline{\rho}$ le plus haut possible. Ces deux paramètres sont directement reliés au nombre de catégories créées. Ainsi, lorsque les valeurs de ces deux paramètres sont élevées, le FAM crée un grand nombre de catégories. Cet effet se perçoit dans la chute des taux de compression (Figure 73.C) lors de l'augmentation des valeurs de ces deux paramètres (Figure 74).

5.5 Conclusion

Dans ce chapitre, nous avons vu que les deux techniques de normalisation testées ont un impact sur les résultats obtenus par les réseaux fuzzy ARTMAP. Nous avons sélectionné la méthode de normalisation MinMax car elle obtient des performances en généralisation semblables ou meilleures, de meilleurs taux de compression et des temps de convergence plus rapides que la technique de normalisation Centrée Réduite.

Après avoir effectué les simulations avec les paramètres standard, les stratégies d'apprentissage standard obtiennent des résultats semblables entre elles, à l'exception de 1EP qui obtient des erreurs en généralisation un peu plus élevées que les trois autres. Le

comportement du FAM semble confirmer que les caractéristiques extraites de la base NIST SD19 possèdent peu ou aucun degré de chevauchement. Ainsi, comme avec les bases DB_{P2} et DB_{CiS}, plus il y aura de patrons dans la base d'entraînement, meilleurs seront les résultats en généralisation.

L'impact de la polarité du MatchTracking a été évalué et, bien que cette dernière n'ait que peu d'impact sur les performances en généralisation, elle influence les taux de compression ainsi que les temps de convergence de la base NIST SD19.

Malheureusement, lors de l'utilisation de la stratégie d'apprentissage spécialisée PSO(1EP), nous n'avons pu effectuer toutes les simulations avec la plage complète du paramètre $\overline{\rho}$ dû au manque de puissance de calcul. Malgré ce fait, les résultats obtenus montrent une amélioration des performances du FAM, obtenant des performances en généralisation généralement meilleures ou égale au 1NN sur la base de données réelles NIST SD19. En réduisant la plage des valeurs du paramètre $\overline{\rho}$, nous avons remarqué une diminution des performances en généralisation des réseaux FAM dépassé un certain nombre de patrons d'entraînement comparativement à l'utilisation de la plage complète.

Deux erreurs en généralisation obtenues lors de l'utilisation de la plage complète du paramètre $\overline{\rho}$ sont supérieures au 1NN, soit avec 52 860 et 150 000 patrons par classe. Ceci est du au fait que nous avons réussi à effectuer seulement deux des quatre itérations PSO avec la base NIST SD19, et ce pour une seule des 10 répétitions. En effectuant les quatre itérations PSO ces résultats rejoindront ceux obtenus avec 1NN.

Finalement, le plus important est qu'il a été démontré que l'optimisation des paramètres permet de maximiser la performance en généralisation des réseaux fuzzy ARTMAP avec la base de données réelles. Ceci démontre, encore une fois, l'importance de la valeur des paramètres lors de l'apprentissage des réseaux fuzzy ARTMAP ainsi que de l'importance de leur optimisation par les stratégies d'apprentissage spécialisées.

CONCLUSION

Dans ce mémoire, nous avons étudié les divers comportements des réseaux fuzzy ARTMAP dans le but de développer une stratégie d'apprentissage spécialisée pour ce type de réseau. La performance des réseaux FAM a été caractérisée sur des bases de données synthétiques et réelles, en observant les effets des stratégies d'apprentissage (qui régissent le nombre d'époques d'entraînement), de la taille de la base d'entraînement, de la technique de normalisation, de la polarité du MatchTracking, et de l'influence des valeurs des quatre paramètres internes FAM (α, β, ε et $\bar{\rho}$).

Pour améliorer les performances en généralisation du réseau FAM, nous avons développé des stratégies d'apprentissage spécialisées pour FAM. Ces stratégies utilisent l'algorithme PSO afin d'optimiser les valeurs des paramètres internes FAM en fonction des performances en généralisation. Les résultats obtenus avec ces stratégies, tant avec les bases de données synthétiques que réelles, montrent l'importance d'une sélection adéquate des paramètres internes des réseaux FAM. Lors de l'optimisation des valeurs des paramètres, le réseau fuzzy ARTMAP tend soit vers un estimateur du centre de masse de chaque classe (tel qu'avec les bases avec chevauchement), soit vers un réseau similaire au 1NN (tel qu'avec les bases sans chevauchement). Pour toutes les bases de données testées, l'utilisation des stratégies d'apprentissage spécialisées, optimisant les paramètres, génèrent des réseaux possédant de meilleures performances en généralisation que lors de l'utilisation des paramètres standard MT- ou MT+ ainsi que des performances généralement meilleures ou égales au *k*NN. Ainsi, les paramètres standard MT- et MT+, qui sont majoritairement utilisés avec FAM, dégradent les performances en généralisation du réseau, et ce, pour toutes les bases testées. La valeur des paramètres internes a donc un impact considérable sur les performances du réseau FAM. Cependant, ces stratégies d'apprentissage spécialisées ne sont pas consistantes avec la philosophie « on-line » de la famille ART, car la valeur des paramètres n'est pas

optimisée séquentiellement. Malgré tout, elle permet d'indiquer les zones de performances optimales pouvant être atteintes par le réseau fuzzy ARTMAP.

Avec les bases possédant un degré de chevauchement, les taux de compression obtenus sont supérieurs à ceux obtenus avec les stratégies standard. Ainsi, l'optimisation des paramètres permet de réduire considérablement le nombre requis de catégories requis pour la classification des bases avec chevauchement, éliminant par le fait même le risque de la prolifération des catégories. Elle permet également de réduire considérablement la dégradation des performances en généralisation due au nombre de patrons d'entraînement. Nous pouvons ainsi utiliser la taille maximale de la base de données d'apprentissage sans risquer d'obtenir une forte dégradation des performances en généralisation telle que subie avec MT-. Cependant, même en optimisant les valeurs des paramètres FAM, il reste une faible erreur de sur-apprentissage présente créée par la taille de la base d'entraînement avec les bases possédant du chevauchement.

Avec les bases de données synthétiques sans chevauchement et la base de données réelles NIST SD19, les performances en généralisation sont également améliorées, principalement lors de l'utilisation de petites tailles des bases d'entraînement. Par contre, les taux de compression de ces bases sont inférieurs à ceux obtenus lors de l'utilisation des paramètres standard MT+ ou MT-.

Lors de l'utilisation des stratégies spécialisées, PSO(1EP) obtient des temps de convergence plus petits que les autres stratégies d'apprentissage spécialisées ($PSO(CONV_P)$, $PSO(CONV_W)$ et PSO(HV)) et ce, pour toutes les simulations effectuées. De plus, PSO(1EP) obtient des performances en généralisation et des taux de compression similaires aux trois autres stratégies d'apprentissage spécialisées et ce, pour toutes les bases synthétiques testées. Il est donc recommandé d'employer uniquement cette stratégie spécialisée afin d'accélérer le temps de convergence.

Les simulations effectuées montrent également que les techniques de normalisation peuvent, dans certains cas, par exemple avec la base de données réelles NIST SD19, influencer les performances des réseaux fuzzy ARTMAP. Cependant, avec les données synthétiques, aucune différence majeure n'a été détectée. Les deux méthodes de création de la structure du chevauchement que nous avons testées obtiennent également des performances similaires. Ainsi, ces deux méthodes créent des bases de données (DB_{μ} et DB_{σ}) de difficultés semblables.

À travers l'ensemble des simulations effectuées, il a été également montré que les réseaux fuzzy ARTMAP peuvent subir des dégradations de performance engendrées par le nombre d'époques d'entraînement ainsi que par la taille de la base d'apprentissage, et ce, pour les bases de données possédant un degré de chevauchement. Lors de l'utilisation des paramètres standard MT- et MT+, la taille de la base d'entraînement est le facteur causant la majorité de l'erreur de sur-apprentissage. Les différentes stratégies d'apprentissage nous ont permis de constater que la dégradation des performances en fonction du nombre d'époques est possible, dans certain cas, avec les réseaux FAM, mais reste faible dans les simulations effectuées. L'augmentation graduelle de la taille de la base d'entraînement nous a permis d'observer que les réseaux FAM entraînés sur les bases de données sans chevauchement ne subissent aucune dégradation de performance due au nombre de patrons d'entraînement.

Nous avons également montré que la polarité du MatchTraking utilisée fait varier les performances obtenues avec les réseaux fuzzy ARTMAP, à la baisse ou à la hausse, et ce, au niveau des performances en généralisation, des taux de compression ainsi que des temps de convergence. Cet aspect nous a permis de démontrer que les paramètres standard MT- et MT+ doivent être utilisés en tenant compte du fait qu'ils ne sont pas optimums pour tous les types de bases de données. Ces paramètres peuvent être utilisés pour obtenir une évaluation rapide d'un problème de classification, tout en sachant que les performances en généralisation seront meilleures une fois leur optimisation effectuée.

Malheureusement, lors des simulations avec la base NIST SD19, nous avons dû réduire la plage de recherche du paramètre $\bar{\rho}_a$ lors de l'optimisation des paramètres. Cette situation est due au manque de puissance de calcul. Malgré cela, il a été démontré que l'optimisation des paramètres est en mesure d'améliorer les performances en généralisation des réseaux fuzzy ARTMAP avec une base de données réelles ainsi qu'avec les bases de données synthétiques, démontrant ainsi la puissance de cette nouvelle méthode d'entraînement pour ce type de réseau.

Lors de l'optimisation des paramètres, la recherche du meilleur réseau s'effectue uniquement en fonction de la performance en généralisation. Si deux réseaux obtiennent la même erreur en généralisation, le réseau ayant le moins de catégories, soit le plus haut taux de compression, est sélectionné. Nous aurions pu faire une optimisation multicritères en tenant compte de l'erreur en généralisation ainsi que du taux de compression lors de l'optimisation PSO. Cette approche permettrait d'obtenir des réseaux ayant des performances similaires à celles que nous avons obtenues, mais avec des taux de compression supérieurs.

Pour valider cette approche un petit test a été effectué en ajoutant une règle dans la mise à jour de la meilleure particule globale (*gbest*) à l'intérieur de l'algorithme PSO. Si un réseau obtient une performance en généralisation assez proche de l'erreur minimale (*gbest*), le réseau ayant le meilleur taux de compression devient le nouveau *gbest*. Si le réseau testé est en dessous de la tolérance comparativement au meilleur réseau (*gbest*) il est rejeté, mais s'il obtient une performance en généralisation supérieure du niveau de tolérance, face à *gbest*, il devient automatiquement le meilleur réseau. Huit tests sont effectués avec huit valeurs de tolérance entre les performances en généralisation des réseaux. Ces tests utilisent la stratégie d'apprentissage PSO(1EP) et la base de données DB_{P2} avec 726 patrons d'entraînement. L'optimisation est répétée 30 fois pour obtenir des résultats moyens.

La Figure 76 présente les erreurs en généralisation des réseaux ainsi que les taux de compression obtenus lors de l'optimisation des paramètres en fonction de la tolérance entre les performances en généralisation.

Lors de l'utilisation des tolérances de 0,1% à 0,4%, on constate que les erreurs en généralisation restent semblables et que les taux de compression ont tendance à augmenter. Ainsi, une approche multicritères pourrait engendrer des réseaux de performances similaires tout en possédant de meilleurs taux de compression.

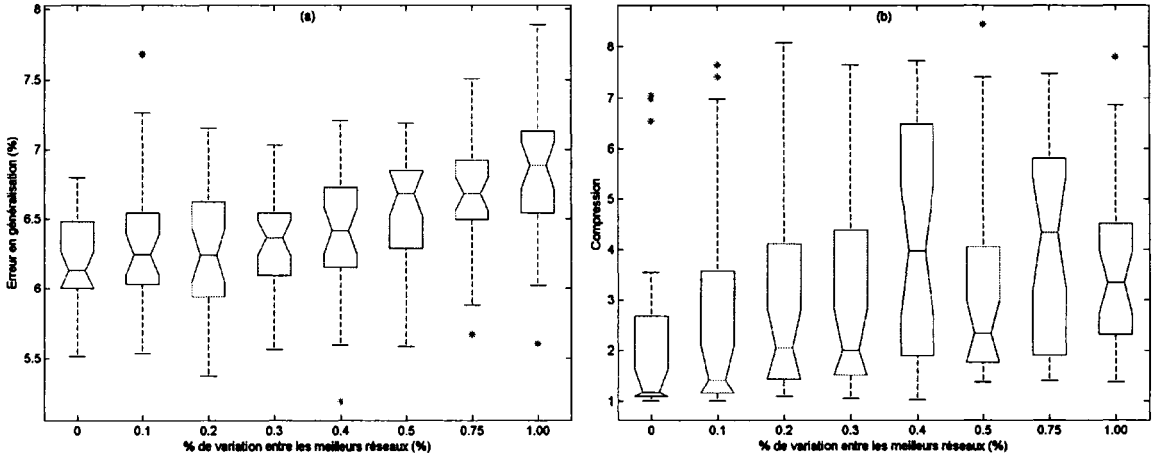


Figure 76 Test PSO utilisant la compression et les erreurs en généralisation
(a) Erreur en généralisation et (b) Taux de compression

Nous aurions également pu comparer les différences d'optimisation entre divers types d'algorithmes d'optimisation et diverses variantes à l'intérieur de chaque algorithme. Nous aurions pu regarder les différences entre l'algorithme PSO synchrone et asynchrone et l'impact des valeurs utilisées à l'intérieur de l'algorithme PSO (nombre de particules, vitesse maximum et minimum, nombre de cycle PSO, etc...) plus en profondeur. Bref, il existe plusieurs algorithmes d'optimisation, chacun ayant ses propres variantes.

Finalement, nous avons réussi à montrer les avantages de notre stratégie d'apprentissage spécialisée qui optimise les paramètres internes FAM lors de l'entraînement des réseaux fuzzy ARTMAP. Nos résultats peuvent encore être améliorés en utilisant une optimisation PSO plus lourde, soit en ayant un plus grand nombre de particules, un nombre d'itérations PSO plus grand, etc. Cependant, notre but n'était pas d'obtenir les meilleures performances possibles du réseau FAM sur les problèmes testés mais bien de démontrer l'efficacité de l'optimisation des paramètres sur les performances en généralisation, les taux de compression ainsi que la dégradation des performances générée par l'utilisation des paramètres standard MT- et MT+. Bref, la valeur des paramètres internes des réseaux fuzzy ARTMAP a un impact considérable sur les performances obtenues et l'utilisation d'une stratégie d'apprentissage spécialisée permettant leur optimisation est nécessaire pour obtenir les pleines performances des réseaux FAM.

ANNEXE 1

Classificateur quadratique bayesien

Le classificateur Bayesien est un classificateur paramétrique purement statistique. Ce type de classifieur repose sur le théorème développé par le mathématicien britannique Thomas Bayes (1702-1761).

Ce théorème est un principe fondamental en théorie des probabilités, issu des travaux de Thomas Bayes et retrouvé ensuite indépendamment par Laplace. Dans son unique article, Bayes cherchait à déterminer la distribution *a posteriori* de la probabilité **P()** d'une loi binomiale. Ses travaux ont été édités et présentés à titre posthume en 1763 [36]. Les résultats de Bayes ont été redémontrés en 1774 par le mathématicien français Laplace [37] qui n'était apparemment pas au fait du travail de Bayes.

Cette annexe présente la théorie du classificateur Bayesien [33]. Les fondements théoriques sont tout d'abord introduits pour un problème comportant deux classes à une seule dimension. Puis, la notion de distribution normale multivariée est présentée.

Fondement

La règle de décision de ce classificateur est basée sur la probabilité d'occurrence (A.1).

$$\begin{aligned} P(w_1 \mid \mathbf{X}) > P(w_2 \mid \mathbf{X}) &\rightarrow w_1 \\ P(w_1 \mid \mathbf{X}) < P(w_2 \mid \mathbf{X}) &\rightarrow w_2 \end{aligned} \qquad \text{(A.1)}$$

Où : **X** est le vecteur d'observation,

w_i fait référence à la classe i,

$P(w_i \mid \mathbf{X})$ est la probabilité *a posteriori* de w_i sachant **X**

L'équation (A.1) indique que si la probabilité que **X** appartienne à la classe w_1 est plus grande que la probabilité qu'il appartienne à la classe w_2, alors **X** est associé à la classe w_1. La probabilité *a posteriori* $P(w_i \mid \mathbf{X})$ est définie par l'équation (A.2).

$$P(w_i \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid w_i) \cdot P(w_i)}{P(\mathbf{X})} \tag{A.2}$$

Où : $P(\mathbf{X})$ est la probabilité *a priori* de **X** telle que défini en (A.3)

$P(w_i)$ est la probabilité *a priori* de la classe w_i

$P(\mathbf{X} \mid w_i)$ est la fonction de vraisemblance de w_i pour **X** connu

$$P(\mathbf{X}) = \sum_{i=1}^{N} P(\mathbf{X} \mid w_i) \cdot P(w_i) \tag{A.3}$$

Où : N est le nombre de classe w

En substituant l'équation (A.2) dans l'équation (A.1) on peut réécrire la règle de décision du classificateur Bayesien basée sur la probabilité d'occurrence sous la forme :

$$\begin{aligned} P(\mathbf{X} \mid w_1) \cdot P(w_1) > P(\mathbf{X} \mid w_2) \cdot P(w_2) \rightarrow w_1 \\ P(\mathbf{X} \mid w_1) \cdot P(w_1) < P(\mathbf{X} \mid w_2) \cdot P(w_2) \rightarrow w_2 \end{aligned} \tag{A.4}$$

Si les probabilités *a priori* $P(w_i)$ des classes w_i sont toutes égales, l'équation (A.4) peut être simplifiée comme le montre l'équation (A.5).

$$\begin{aligned} P(\mathbf{X} \mid w_1) > P(\mathbf{X} \mid w_2) \rightarrow w_1 \\ P(\mathbf{X} \mid w_1) < P(\mathbf{X} \mid w_2) \rightarrow w_2 \end{aligned} \tag{A.5}$$

Pour un vecteur d'observation **X** à une 1 dimension, le théorème de Bayes démontre que la probabilité d'erreur minimale est la surface qui résulte de l'intersection des deux distributions. La figure 77 présente les variations de $P(\mathbf{X} \mid w_i), i = 1,2$, soit deux classes équiprobables, d'une dimension, dont chaque classe respecte une distribution normale. La surface hachurée représente l'erreur minimale, soit l'intersection entre ces deux distributions.

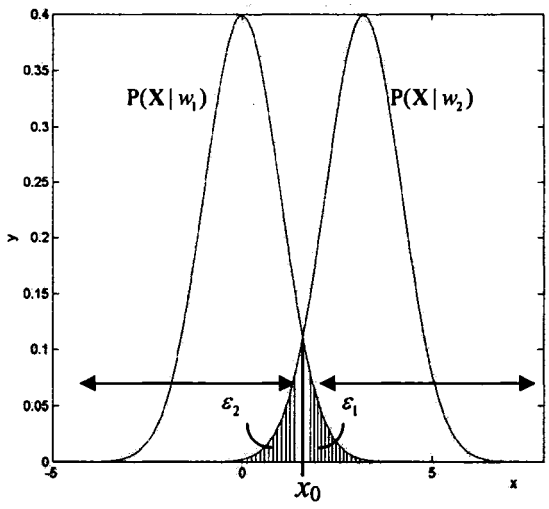


Figure 77 Borne de décision entre deux distributions normales

La règle de décision du classificateur Bayesien fait en sorte que toutes les valeurs de **X** contenues en R1 sont classifiées comme appartenant à la classe w_1 et que toutes les valeurs de **X** contenues en R2 comme appartenant à la classe w_2. Cependant, il est évident en regardant la figure 77 qu'une erreur de classification est inévitable. Il existe une probabilité qu'une donnée présente dans la région R1 appartienne en fait à la classe w_2. Dans le cas présent, la surface hachurée est égale à la probabilité d'erreur totale P_e tel que :

$$2 \cdot P_e = \int_{-\infty}^{x_0} P(\mathbf{X} \mid w_2) dx + \int_{x_0}^{\infty} P(\mathbf{X} \mid w_1) dx \qquad (A.6)$$

Il est possible de démontrer mathématiquement que, sur un tel problème de classification, le classificateur Bayesien obtiendra la plus petite probabilité d'erreur possible.

Distribution normale

Pour trouver la plus petite probabilité d'erreur possible, le classificateur Bayesien doit trouver la frontière de décision optimale entre les classes. Pour un problème de deux

classes, la frontière de décision optimale est obtenue en résolvant l'équation (A.7). Le principe reste semblable pour un problème ayant plus de deux classes.

$$P(\mathbf{X} \mid w_1) = P(\mathbf{X} \mid w_2) \tag{A.7}$$

Pour résoudre cette équation (A.7), le classificateur Bayesien doit estimer la forme de la distribution des données de chaque classe. La forme de distribution la plus souvent utilisée est celle respectant une loi normale (Gaussienne).

La loi normale multivariée décrivant la distribution de N classes w_i de l dimensions est décrite par l'équation (A.8).

$$P(\mathbf{X} \mid w_i) = \frac{1}{\sqrt{2\pi} \cdot |\Sigma_i|^{\frac{1}{2}}} \cdot e^{\left(-\frac{1}{2}(x-\mu_i)^T \cdot \Sigma_i^{-1} \cdot (x-\mu_i)\right)}, i = 1, \ldots, N \tag{A.8}$$

Où : μ_i est la moyenne de la classe w_i

Σ_i est la matrice l x l de covariance

$|\Sigma_i|$ est le déterminant de la matrice de covariance

La matrice de covariance Σ_i est définie comme étant :

$$\Sigma_i = E\left[(x-\mu_i)(x-\mu_i)^T\right] \tag{A.9}$$

Où : $E[\cdot]$ est la valeur moyenne

Étant donné la forme exponentielle de l'équation (A.8), il est préférable de travailler avec la fonction discriminante (A.10) utilisant le logarithme.

$$g_i(x) = \ln(P(\mathbf{X} \mid w_i) \cdot P(w_i)) = \ln(P(\mathbf{X} \mid w_i)) + \ln(P(w_i)) \tag{A.10}$$

En tenant compte de la distribution normale multivariée (A.8), la fonction discriminante (A.10) peut être réécrite sous la forme :

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T \sum_i^{-1} \cdot (x-\mu_i) + \ln(P(w_i)) + c_i \qquad (A.11)$$

Où : c_i est une constante de valeur $c_i = -\frac{l}{2} \cdot \ln(2\pi) - \frac{1}{2}\ln(|\Sigma_i|)$

On peut réécrire l'équation (A.11) sous la forme:

$$g_i(x) = -\frac{1}{2}x^T\Sigma_i^{-1}x + \frac{1}{2}x^T\Sigma_i^{-1}\mu_i - \frac{1}{2}\mu_i^T\Sigma_i^{-1}\mu_i + \frac{1}{2}\mu_i^T\Sigma_i^{-1}x + \ln(P(w_i)) + c_i \qquad (A.12)$$

En général, l'équation (A.12) est une fonction quadratique non linéaire. Supposons le cas d'un problème de deux classes, $l = 2$, ainsi que la matrice de covariance suivante :

$$\Sigma_i = \begin{pmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2 \end{pmatrix} \qquad (A.13)$$

En utilisant cette matrice de covariance, l'équation (A.12) s'écrit alors sous la forme :

$$g_i(x) = -\frac{1}{2\sigma_i^2}(x_1^2 + x_2^2) + \frac{1}{\sigma_i^2}(\mu_{i1}x_1 + \mu_{i2}x_2) - \frac{1}{2\sigma_i^2}(\mu_{i1}^2 + \mu_{i2}^2) + \ln(P(w_i)) + c_i \qquad (A.14)$$

On peut constater que la frontière de décision obtenue lorsque $g_i(x) = g_j(x)$ est de forme quadratique *(elliptique, parabolique, hyperbolique)*. Dans un tel cas, le classificateur Bayesien est un classificateur quadratique dans le sens où la frontière de décision utilisée pour la classification des données est une fonction quadratique. Lors de

l'utilisation de la formule (A.14) pour établir une frontière de décision, le classificateur Bayesien est appelé le classificateur quadratique Bayesien.

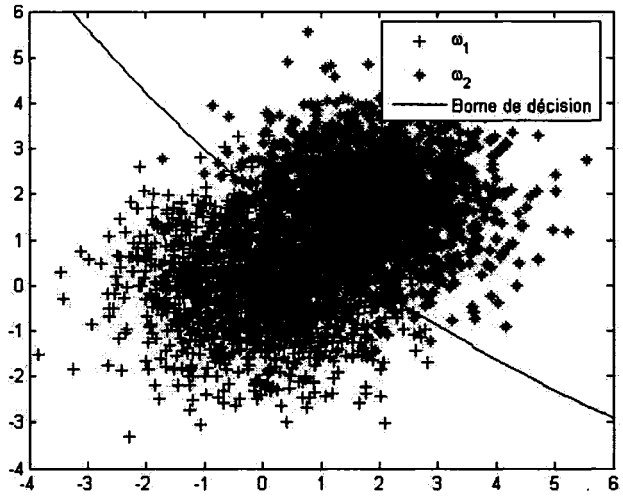


Figure 78 Borne de décision entre deux distributions normales bivariées

Lors de l'utilisation d'un classificateur quadratique Bayesien, une base de référence est utilisée pour calculer la moyenne μ et la matrice de covariance Σ pour chaque classe. À partir de ces valeurs, la théorème de Bayes est appliqué pour calculer la probabilité d'erreur minimale.

ANNEXE 2

La règle du *k* plus proches voisins (*k*NN)

Le classificateur utilisant la règle des k plus proches voisins (ou kNN pour *"k Nearest Networks"* [33]) a été proposé en 1967 par Cover et Hart [24]. Depuis, ce classificateur a connu plusieurs améliorations. Un ouvrage réalisé par Dasarathy [25] regroupe un grand nombre d'articles présentant un historique des méthodes utilisant le kNN ainsi que ses diverses évolutions.

L'algorithme du kNN est une technique paramétrique qui nécessite l'utilisation d'une base de données d'apprentissage. Cependant, ce réseau ne requiert pas de phase d'entraînement. La base de données d'apprentissage est utilisée pour représenter la dispersion des classes contenues dans l'espace des données. Lorsqu'une observation $\boldsymbol{X}$ est présentée au réseau, le classificateur effectue une mesure de distance, souvent appelée entropie, entre l'observation $\boldsymbol{X}$ et tous les points contenus dans la base de référence (base d'apprentissage). Puis les k plus proches voisins de l'observation $\boldsymbol{X}$ votent pour l'attribution de la classe de $\boldsymbol{X}$. Ainsi, pour un problème à deux classes (w_1 et w_2), si $k = 3$, les 3 plus proches voisins de l'observation $\boldsymbol{X}$ sont sélectionnés comme votants. Si deux de ces trois voisins proviennent de la classe w_1 la classe attribuée à l'observation $\boldsymbol{X}$ sera w_1. Pour garantir qu'il n'y aura jamais d'égalité dans les votes, k doit être égal à un multiple du nombre de classe plus un (B.1).

$$k = 1 + \alpha \cdot \|W\| \tag{B.1}$$

Où : k est le nombre de plus proches voisins utilisés pour le vote

α est un nombre réel $\{0, 1, 2, \ldots\}$

$\|W\|$ est le nombre de classe contenues dans le problème de classification

L'équation de la distance Euclidienne (B.2) est généralement utilisée pour le calcul de la distance entre l'observation à classifier et les points contenus dans la base de référence. Par contre, la distance Euclidienne est sensible à la normalisation des données. Il existe

d'autres méthodes pour le calcul de la distance comme l'équation de Mahalanobis utilisant l'inverse de la matrice de covariance ainsi que la distance de Hamming.

$$D_e = \sqrt{\sum_{o<f\leq d} (x_f' - x_f)^2} \tag{B.2}$$

La performance du *k*NN est directement reliée à la taille de la base d'apprentissage. Si la taille de cette base tend vers l'infini, la probabilité d'erreur P(E_{1NN}) d'un classificateur 1NN (k=1) est contenue dans l'intervalle $P(E_b) \leq P(E_{1NN}) \leq 2 \cdot P(E_b)$, où P($E_b$) est la probabilité d'erreur bayesienne. La valeur du paramètre k joue également un rôle important sur la performance de la classification par la règle des plus proches voisins. À mesure que la valeur de k augmente, la probabilité d'erreur *a posteriori* $P(w_i \mid \mathbf{x}) = q(\mathbf{x})$ tend vers sa valeur optimale. Par contre, il faut s'assurer que la distance entre les k points utilisés pour le vote de la classification est relativement proche du point observé A. En général, une faible valeur de k est utilisée lors des problèmes ne possédant aucun degré de chevauchement. Pour des problèmes complexes et/ou avec degré de chevauchement, la valeur du paramètre k doit être ajustée pour chaque problème de classification.

ANNEXE 3

Résultats généraux pour l'ensemble des degrés de chevauchement

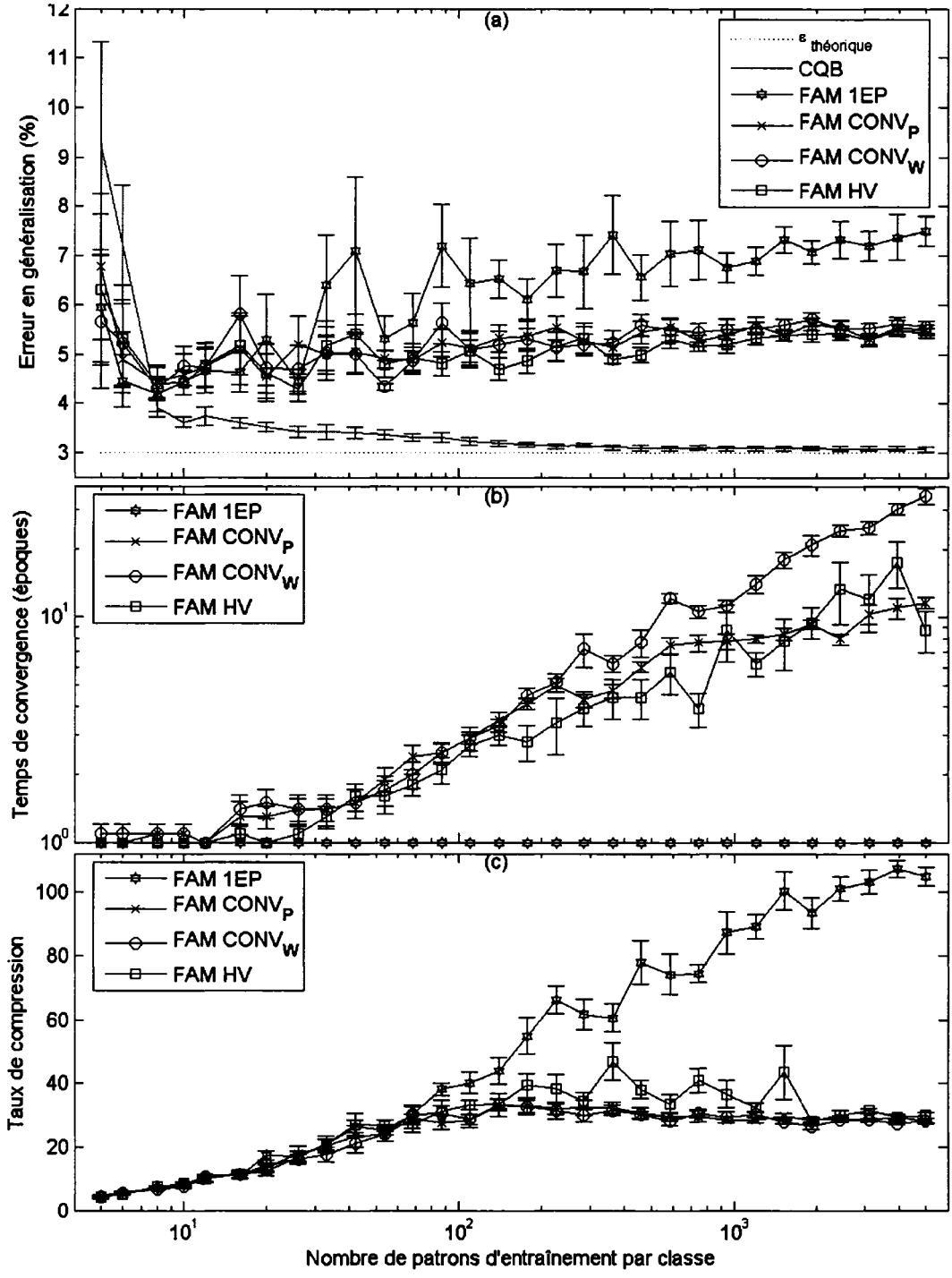


Figure 79 Performances du FAM en fonction de la taille de la base d'apprentissage avec la base DBμ(3%)

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

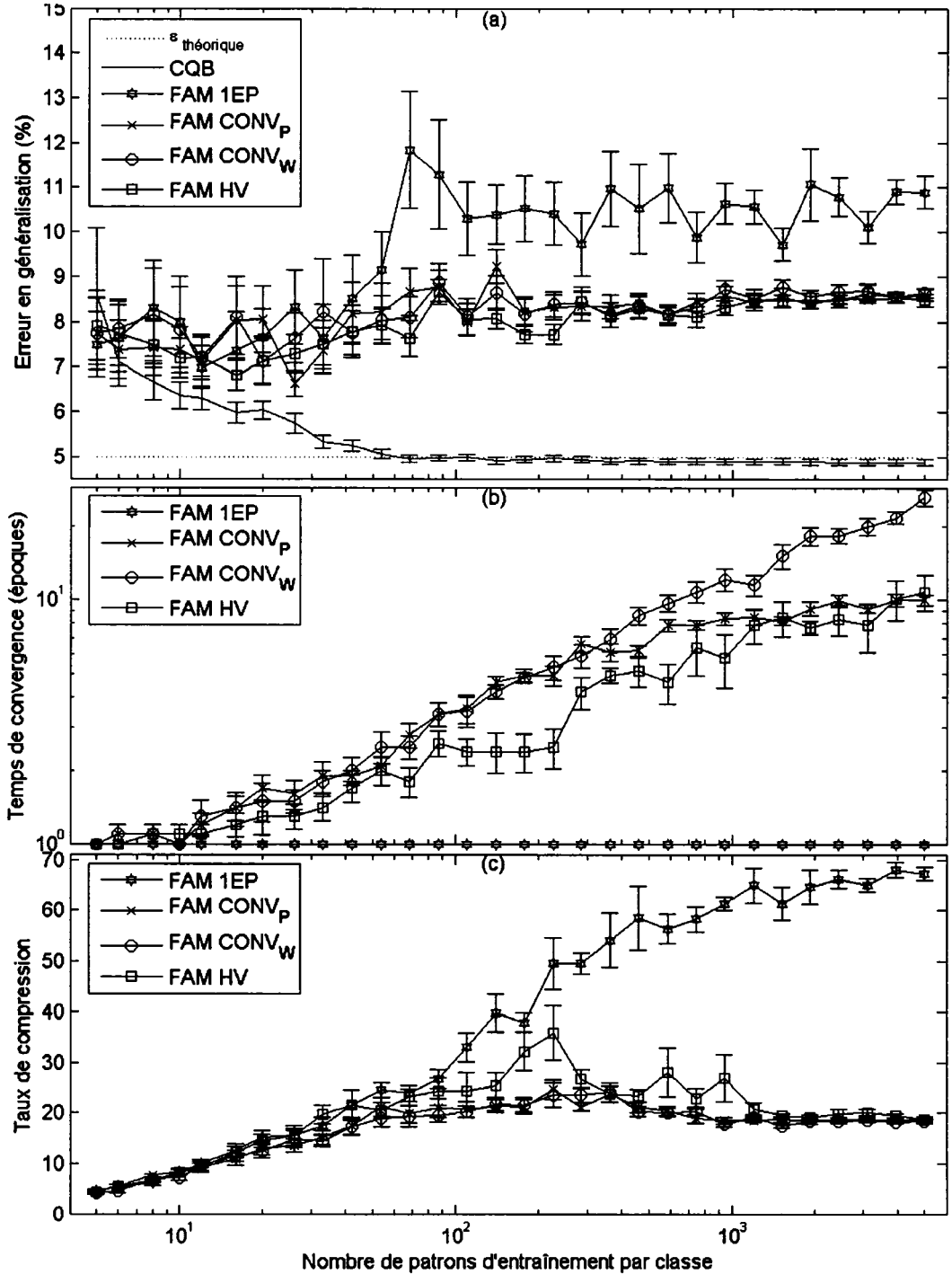


Figure 80 Performances du FAM en fonction de la taille de la base d'apprentissage avec la base DBμ(5%)

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

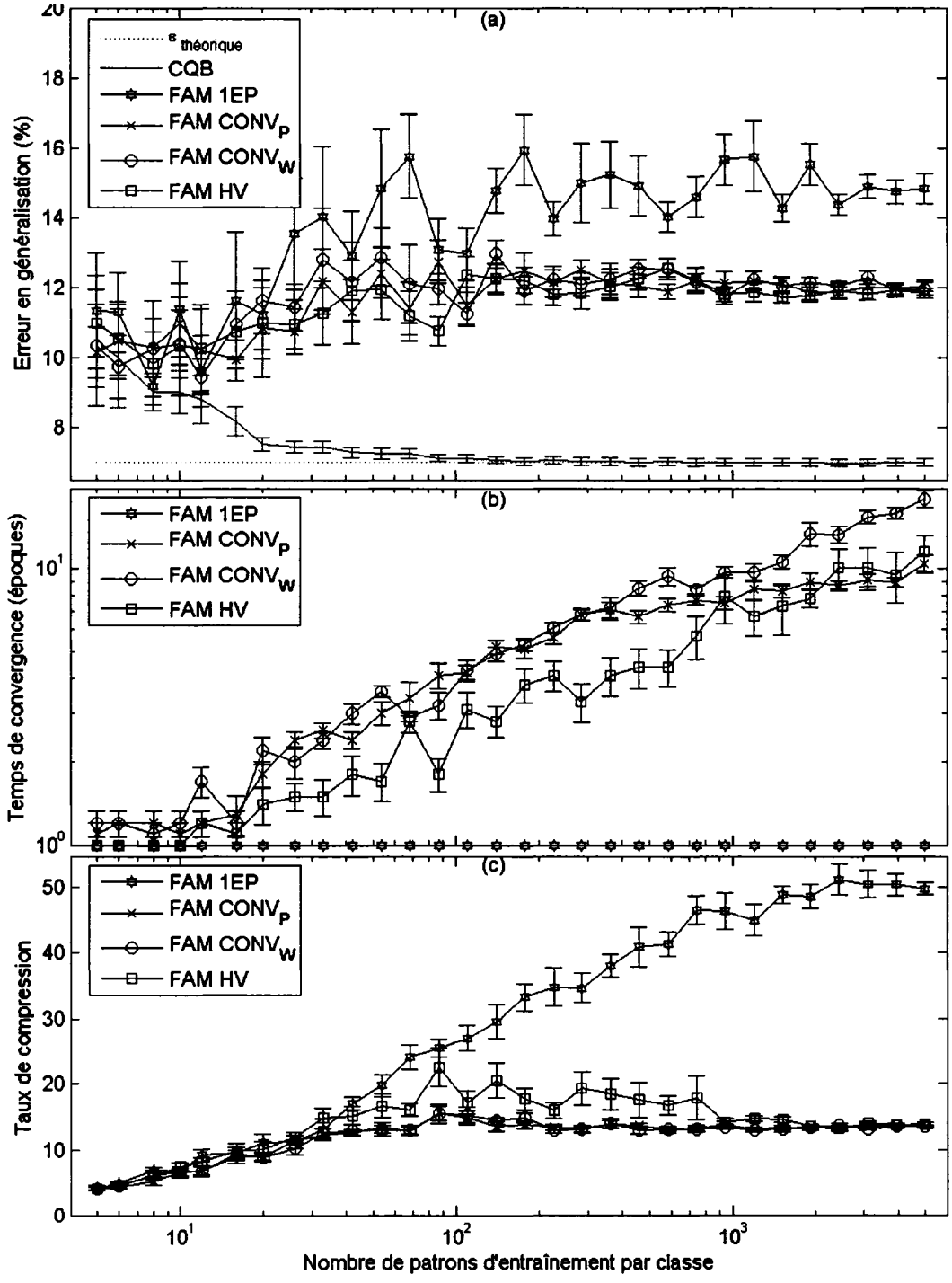


Figure 81 Performances du FAM en fonction de la taille de la base d'apprentissage avec la base DBμ(7%)

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

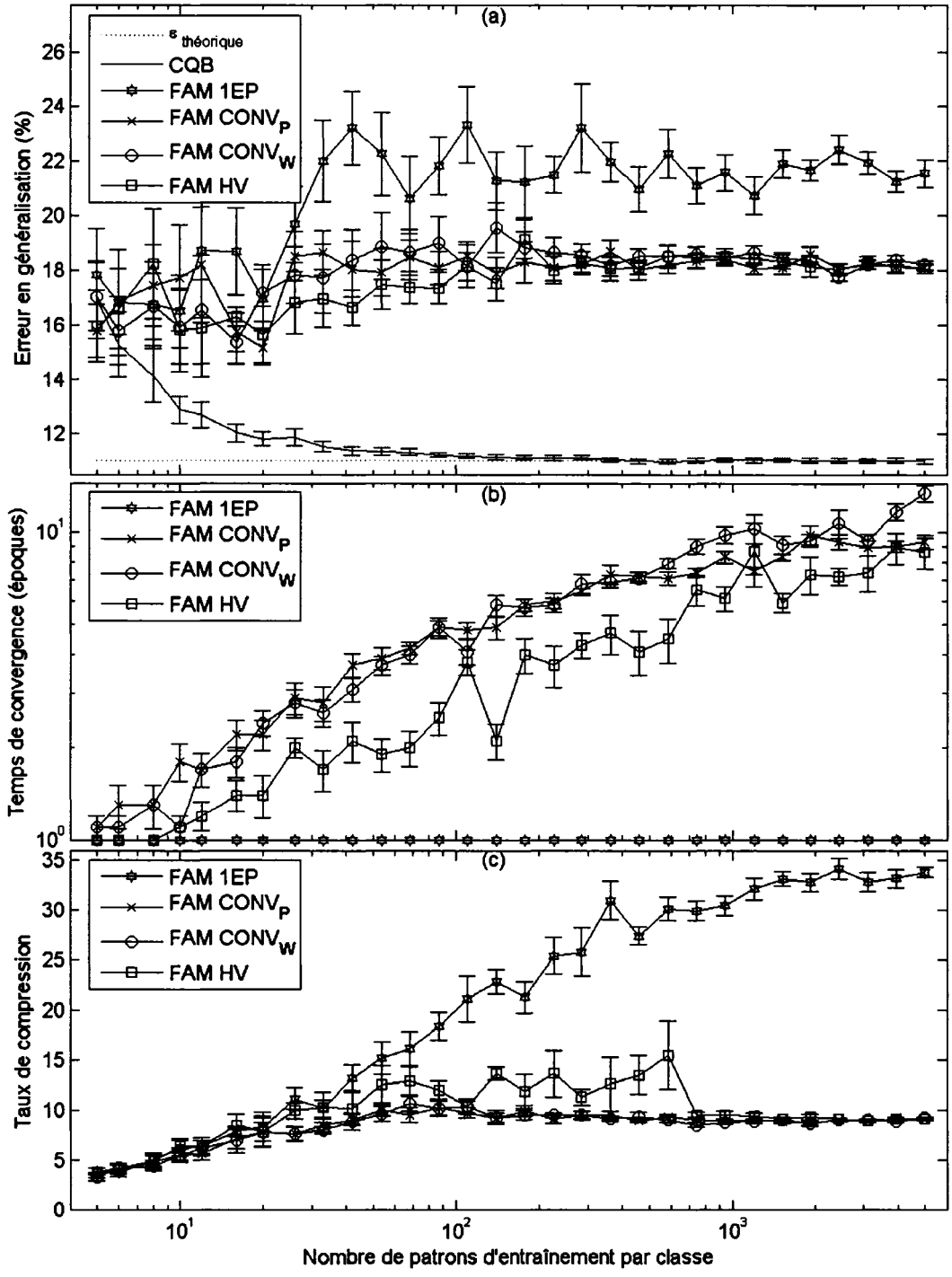


Figure 82 Performances du FAM en fonction de la taille de la base d'apprentissage avec la base DBμ(11%)

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

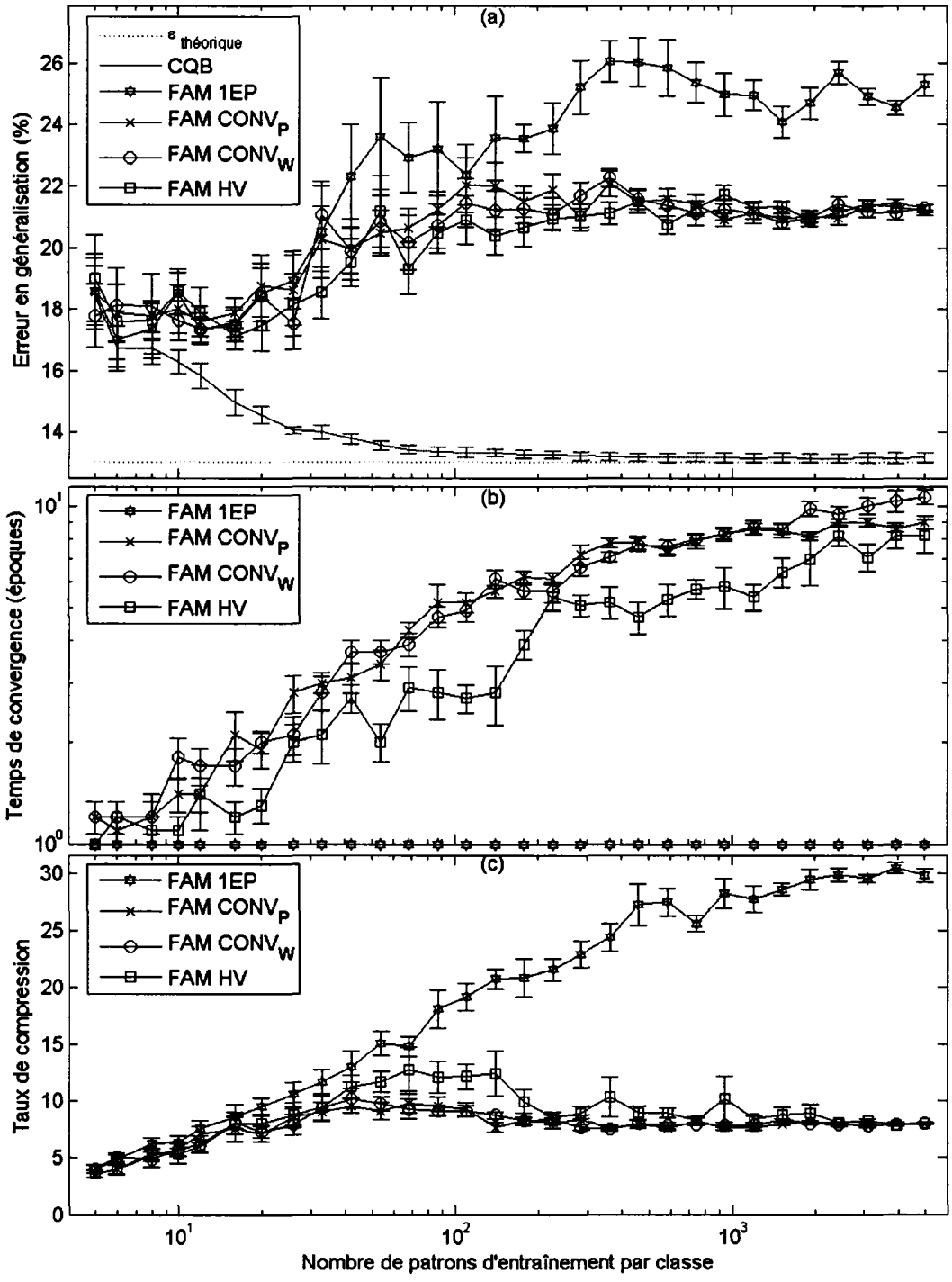


Figure 83 Performances du FAM en fonction de la taille de la base d'apprentissage avec la base DBμ(13%)

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

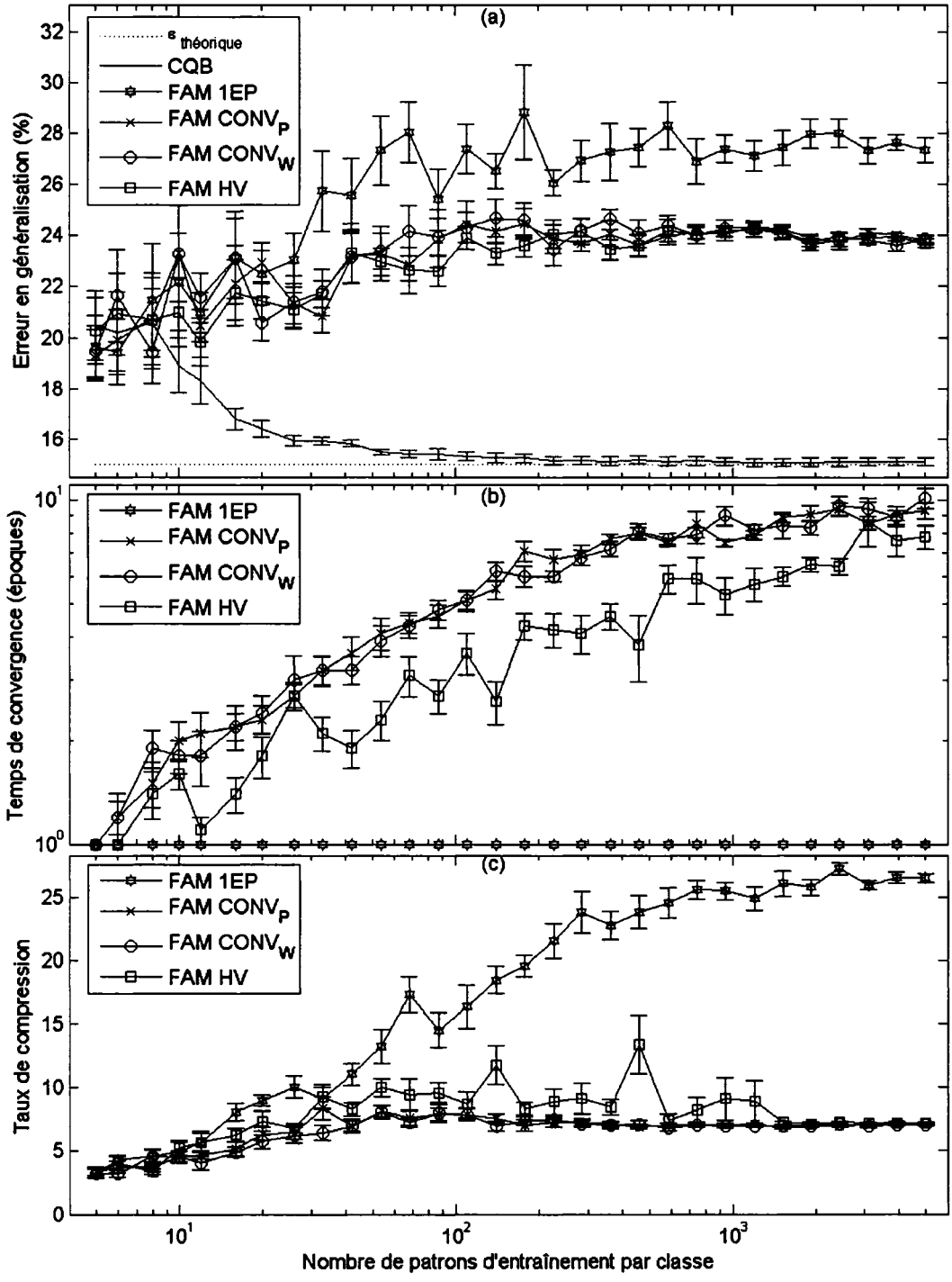


Figure 84 Performances du FAM en fonction de la taille de la base d'apprentissage avec la base DBμ(15%)

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

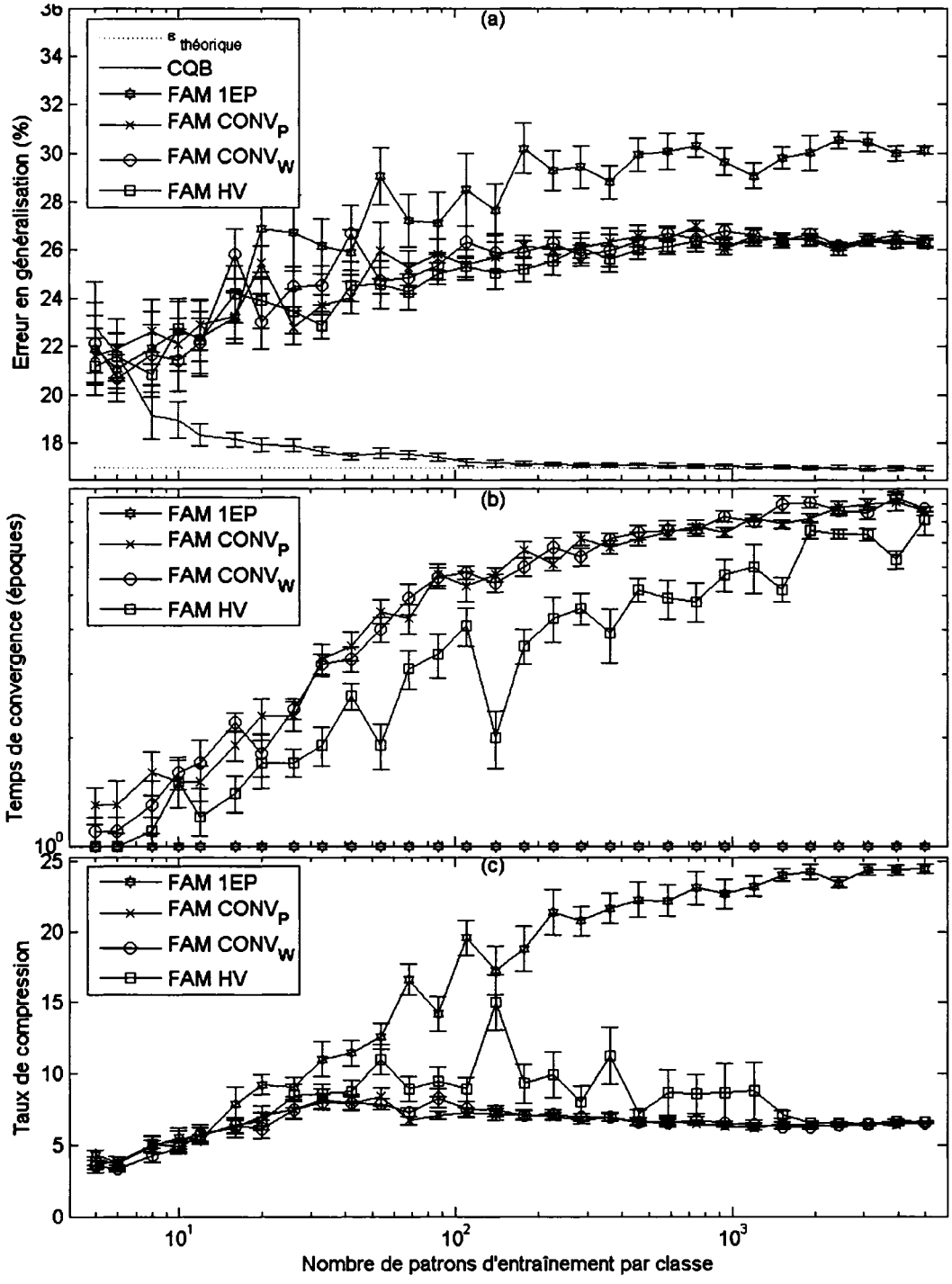


Figure 85 Performances du FAM en fonction de la taille de la base d'apprentissage avec la base DBμ(17%)

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

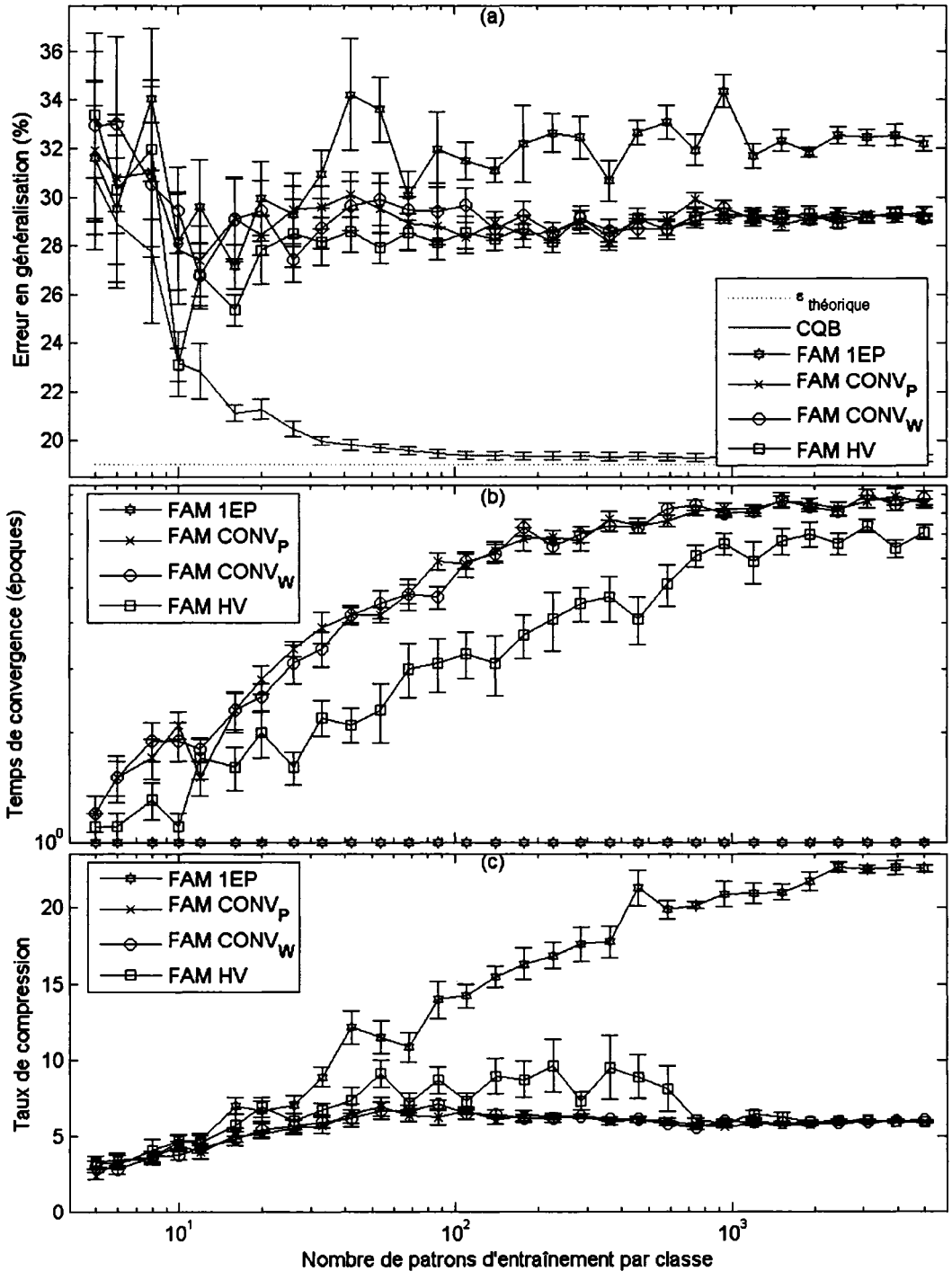


Figure 86 Performances du FAM en fonction de la taille de la base d'apprentissage avec la base DBμ(19%)

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

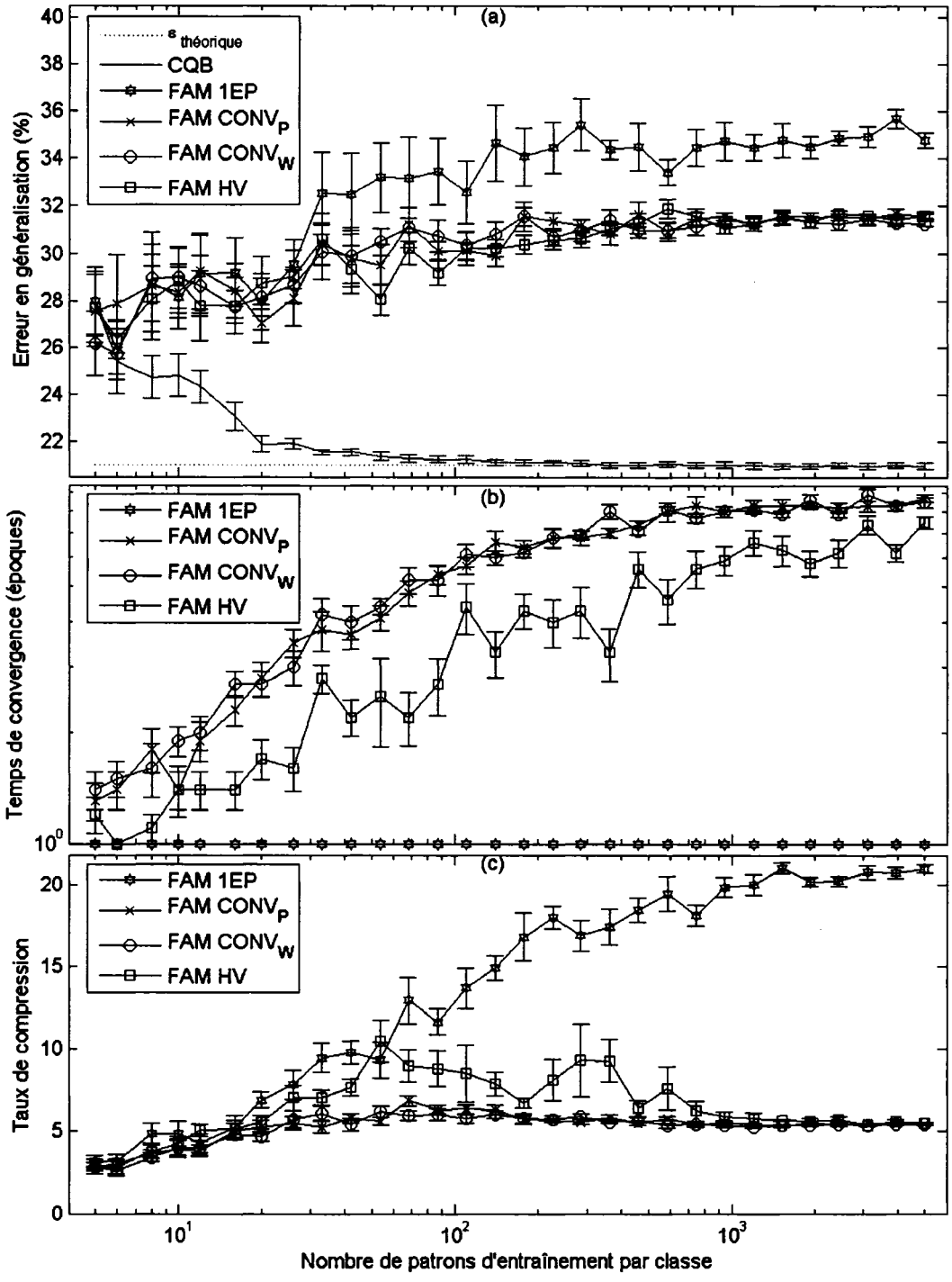


Figure 87 Performances du FAM en fonction de la taille de la base d'apprentissage avec la base DBμ(21%)

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

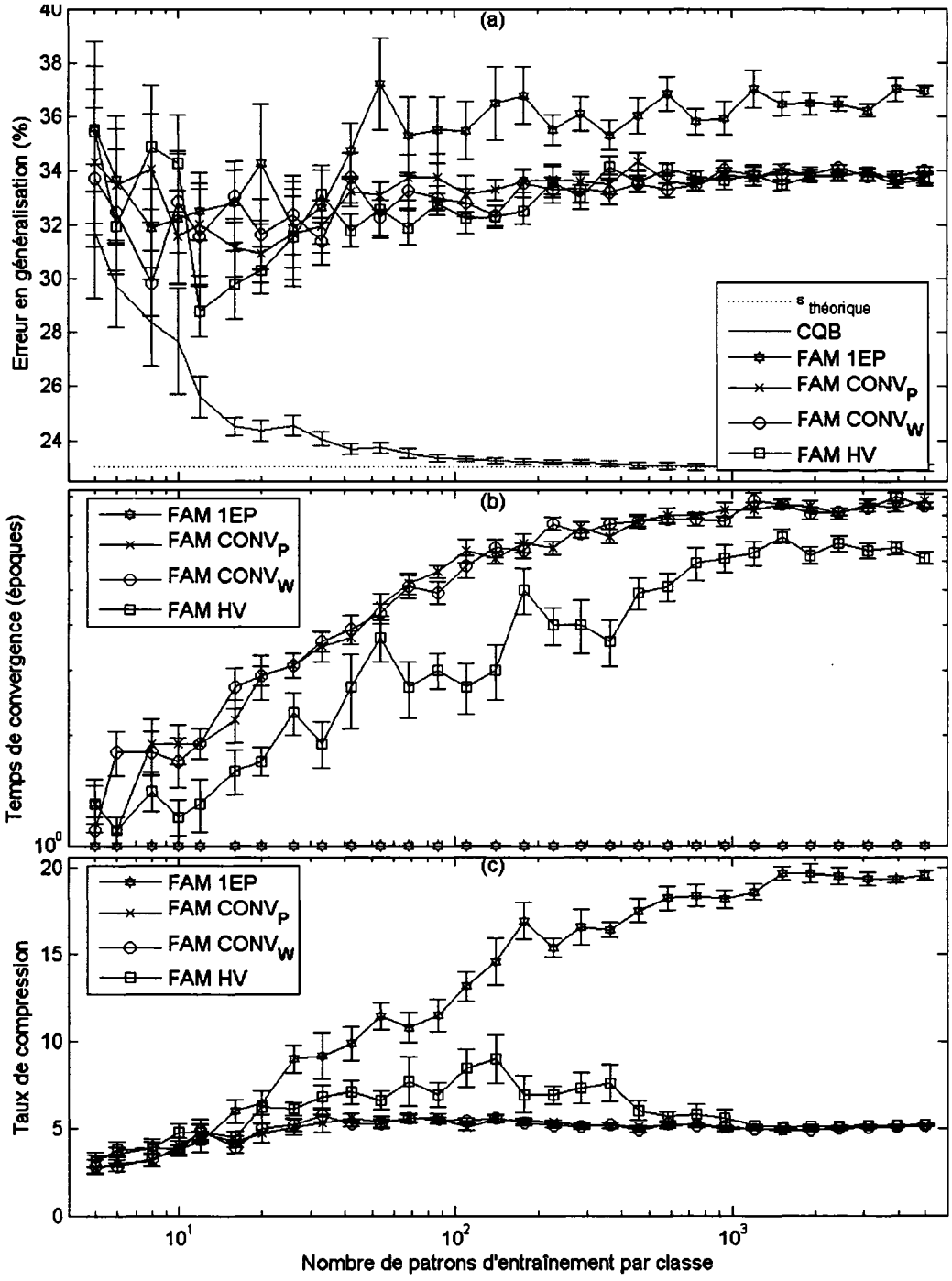


Figure 88 Performances du FAM en fonction de la taille de la base d'apprentissage avec la base DBμ(23%)

(a) Erreur en généralisation, (b) temps de convergence et (c) taux de compression

ANNEXE 4

Effets de la structure du chevauchement des données synthétiques

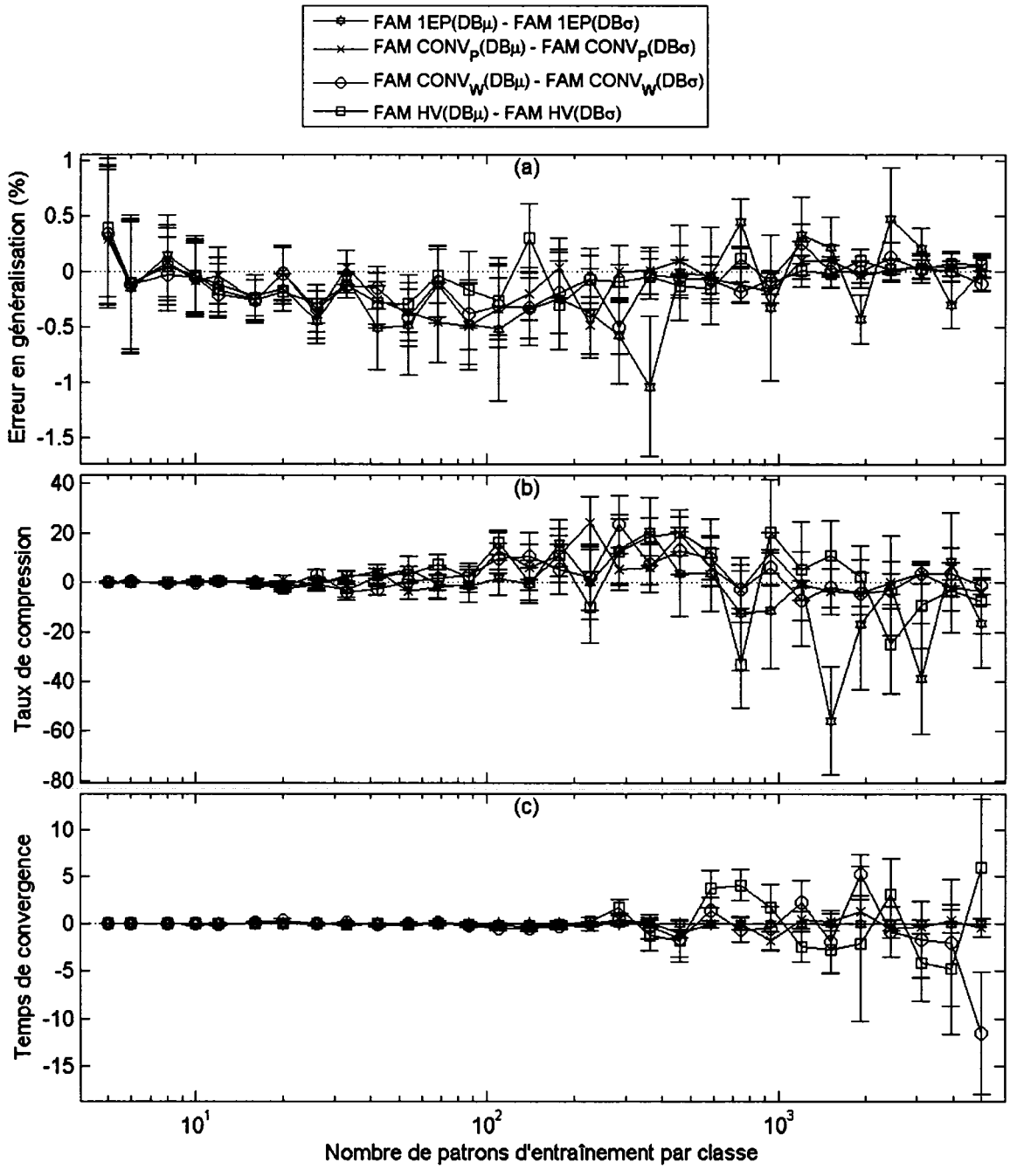


Figure 89 Différence des performances du FAM entre DBμ(1%) et DB$_\sigma$(1%)
(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

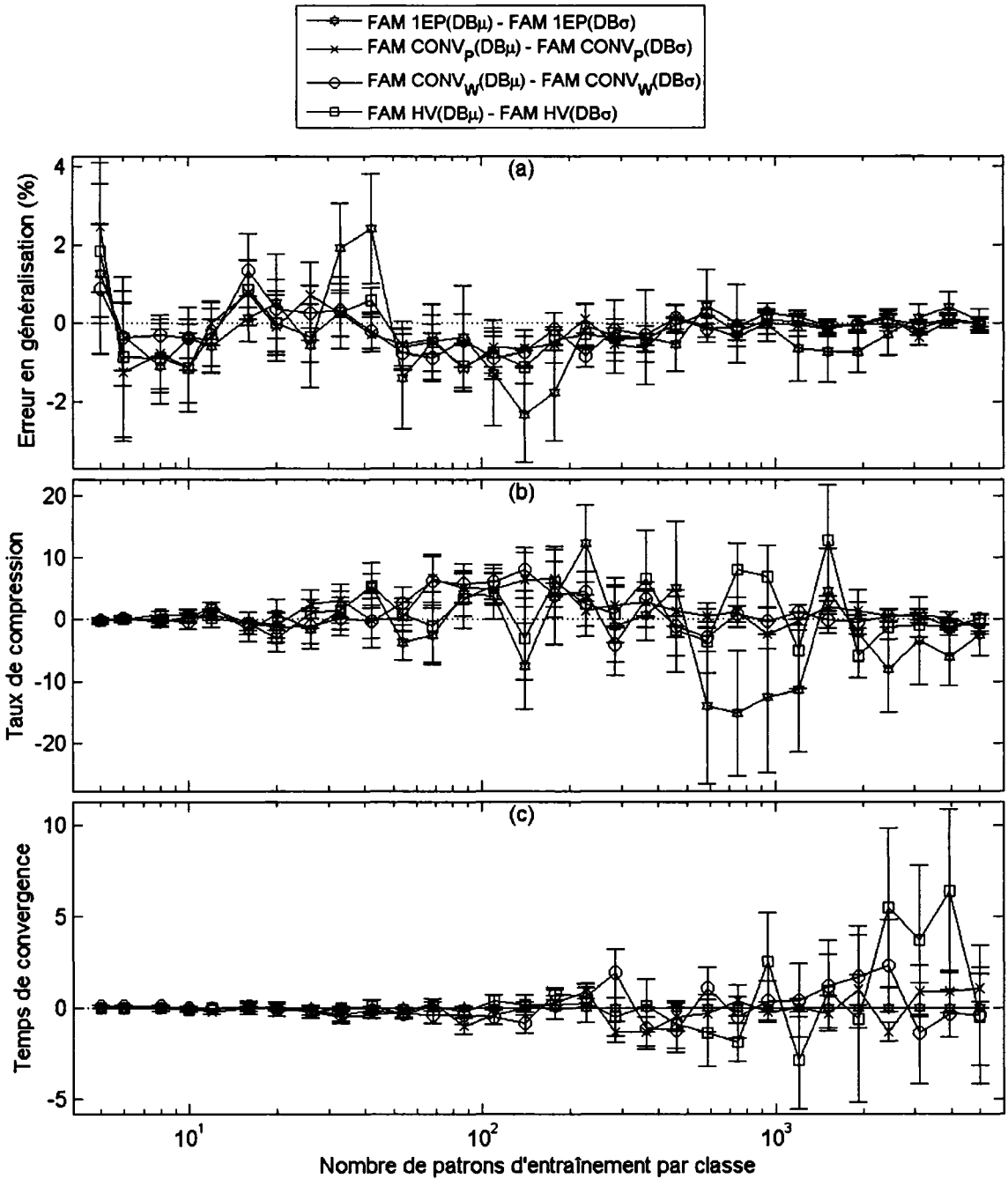


Figure 90 Différence des performances du FAM entre DBμ(3%) et DB$_\sigma$(3%)
(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

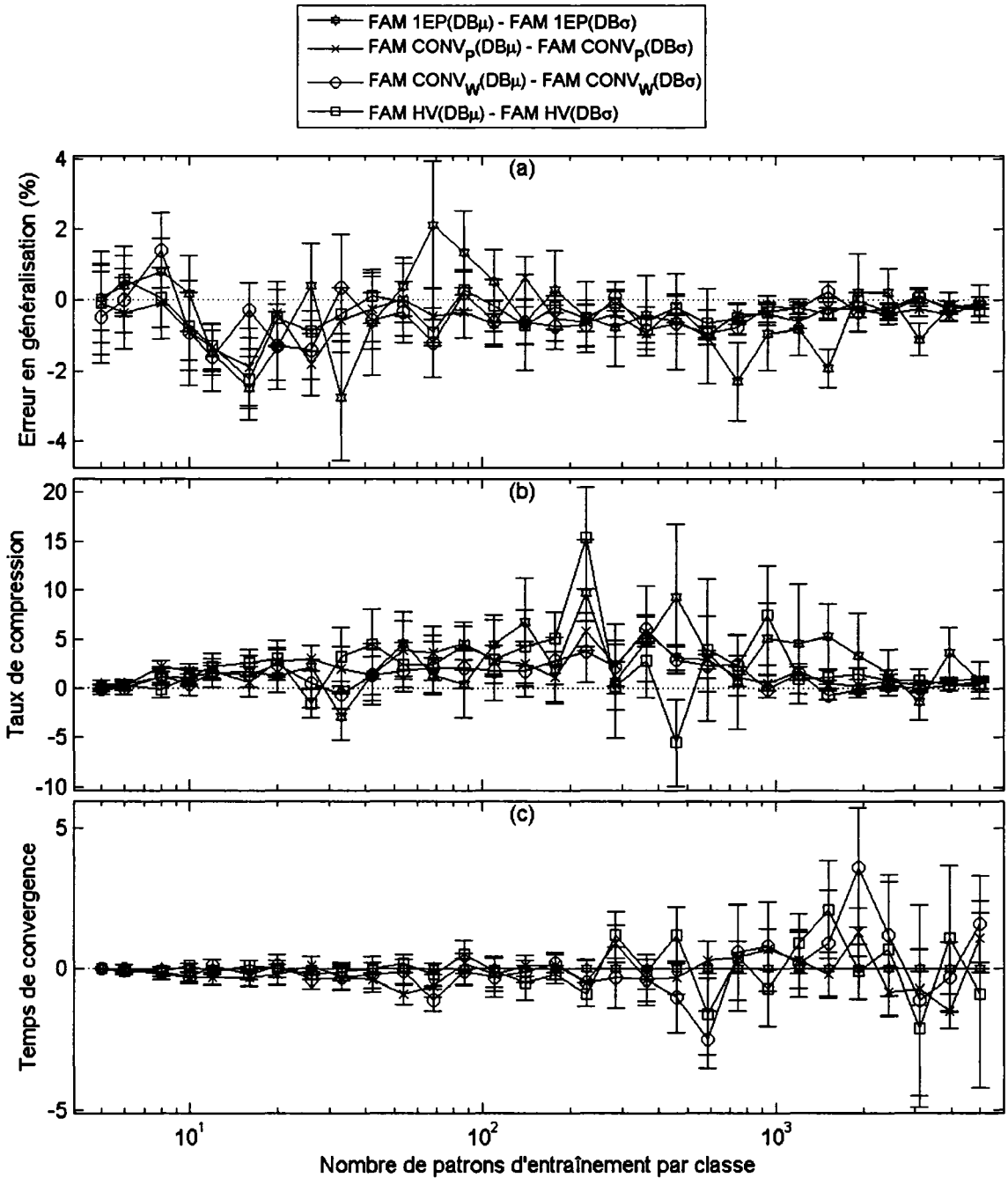


Figure 91 Différence des performances du FAM entre DBμ(5%) et DB$_\sigma$(5%)
(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

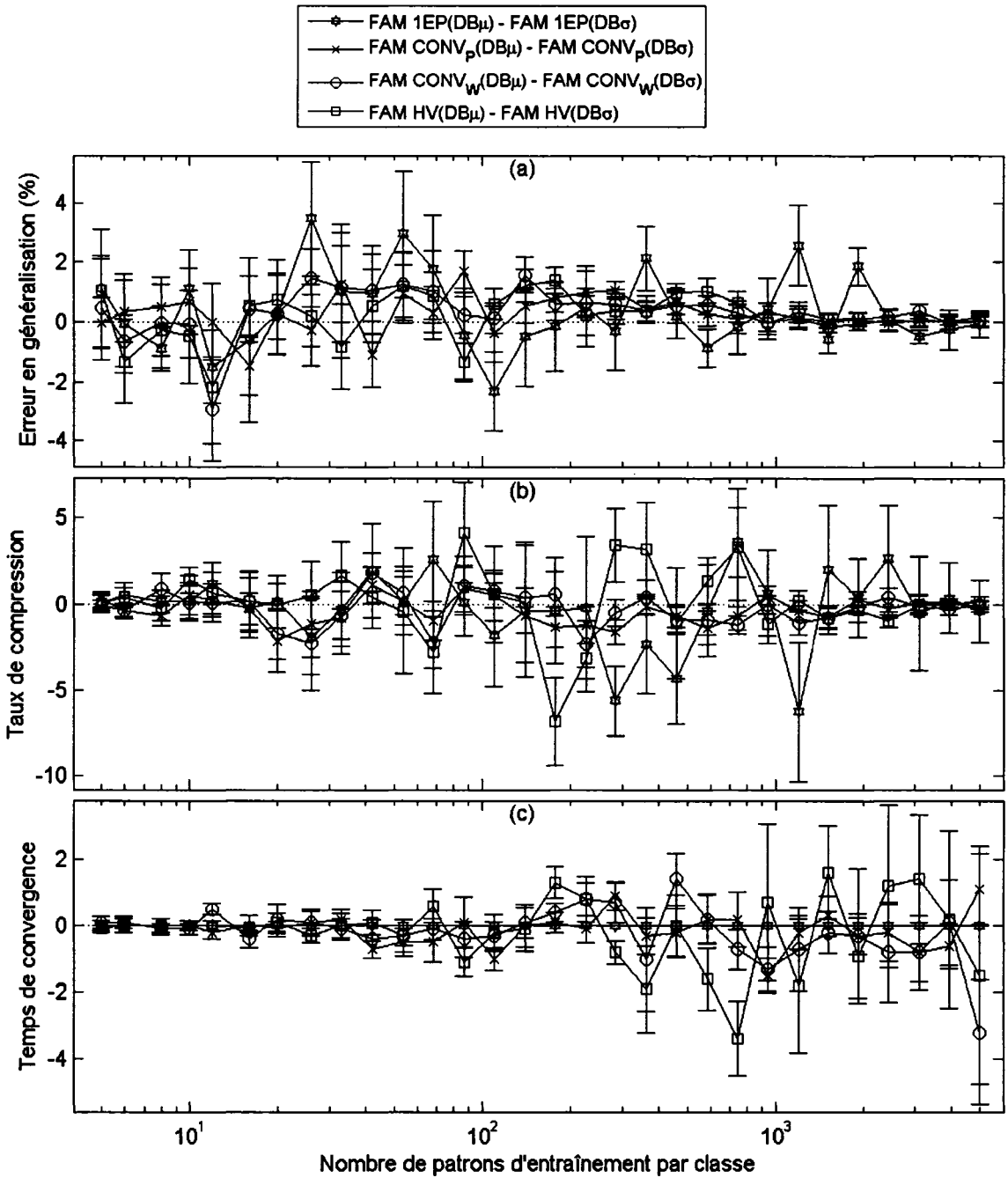


Figure 92 Différence des performances du FAM entre DBμ(7%) et DB$_\sigma$(7%)
(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

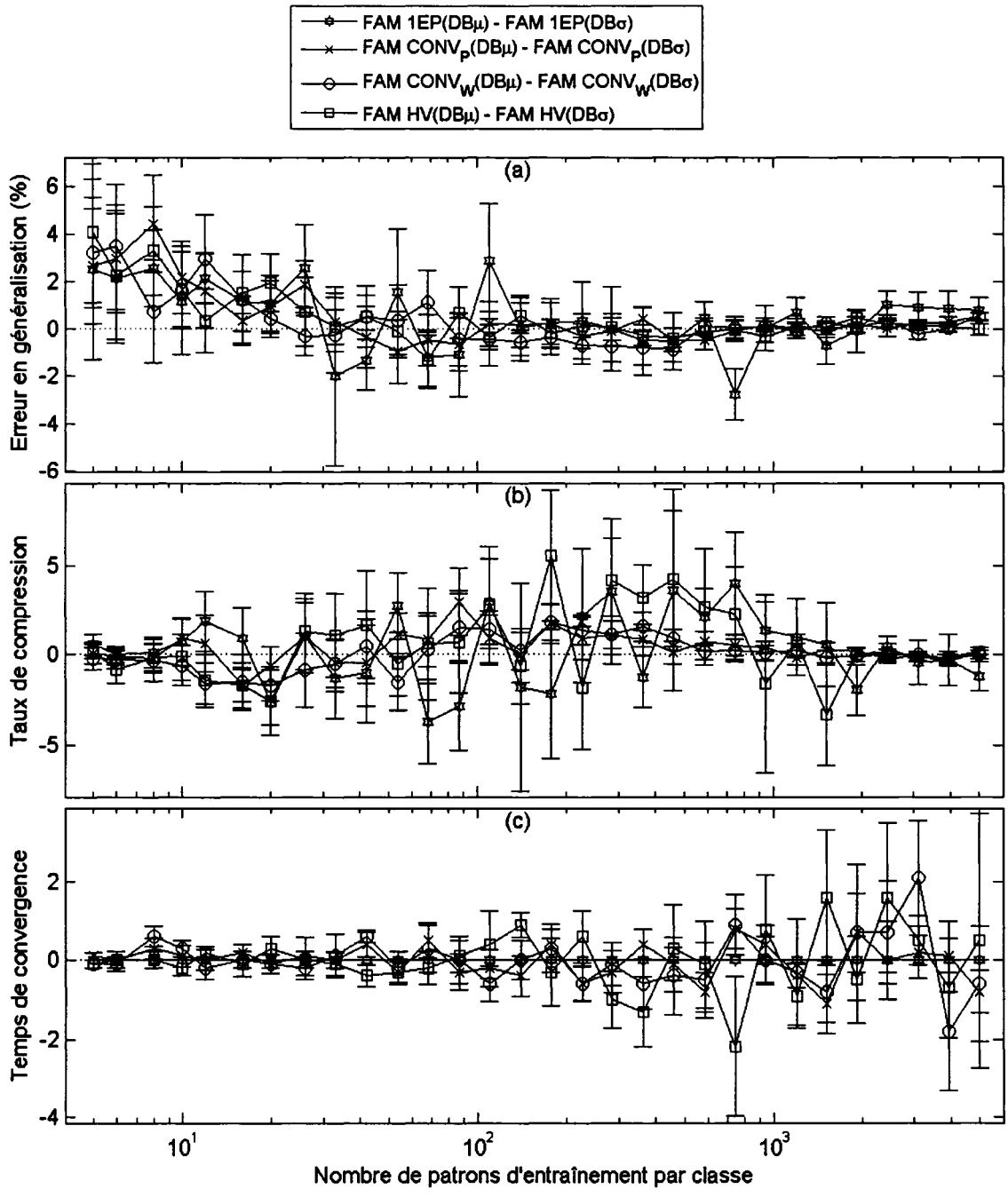


Figure 93 Différence des performances du FAM entre DBμ(9%) et DB$_\sigma$(9%)
(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

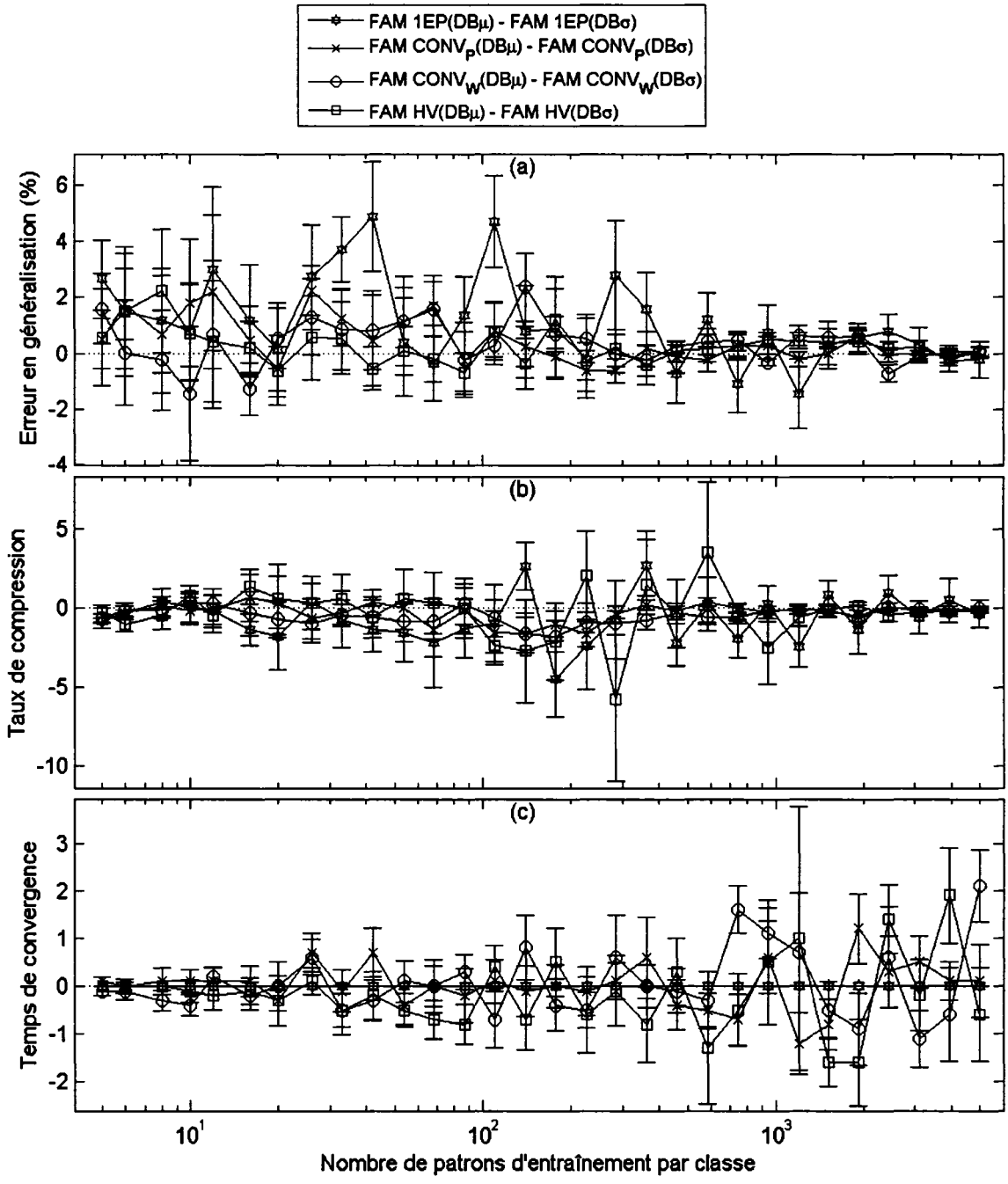


Figure 94 Différence des performances du FAM entre DBμ(11%) et DB$_\sigma$(11%)
(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

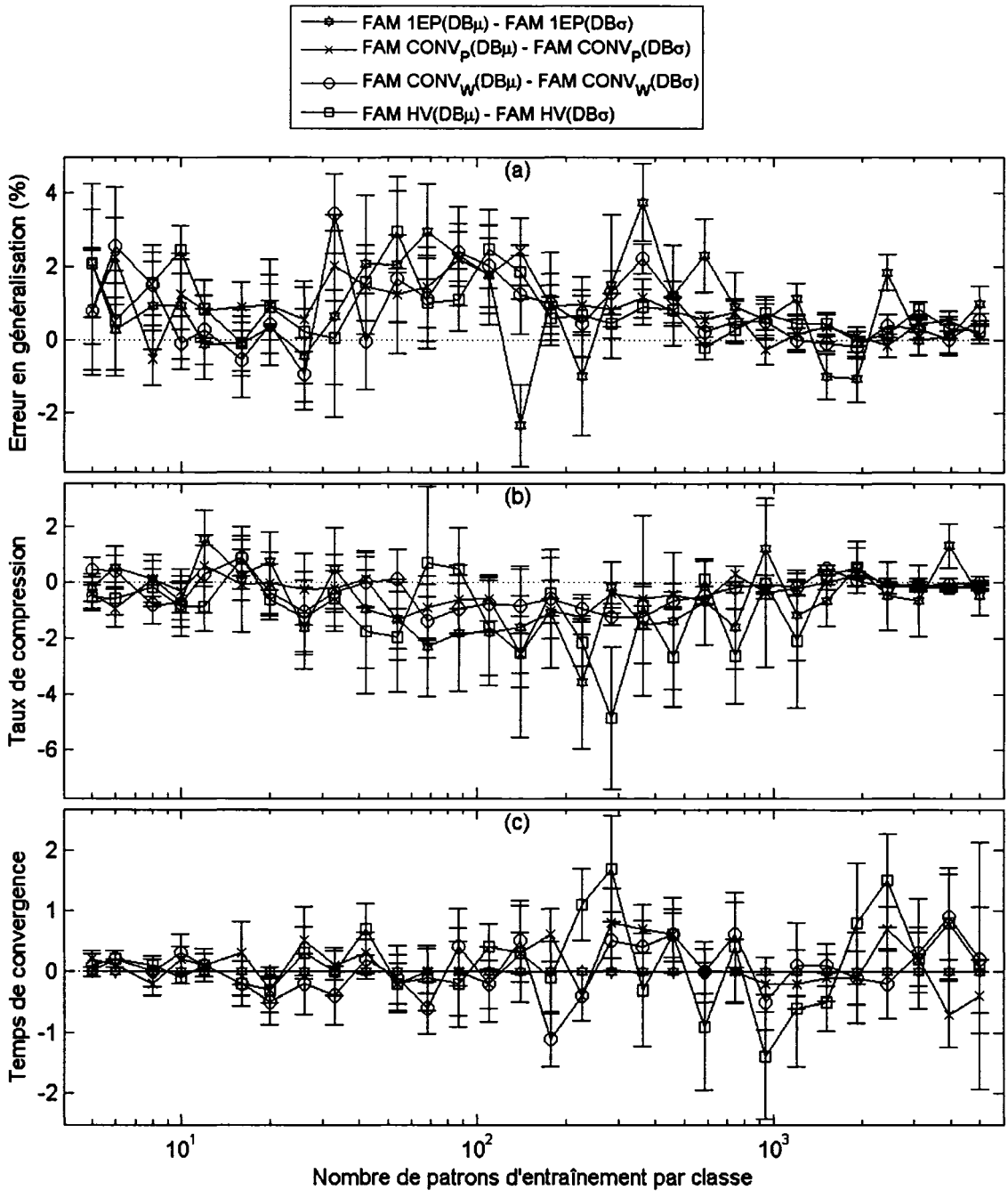


Figure 95 Différence des performances du FAM entre DBμ(13%) et DB$_\sigma$(13%) (a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

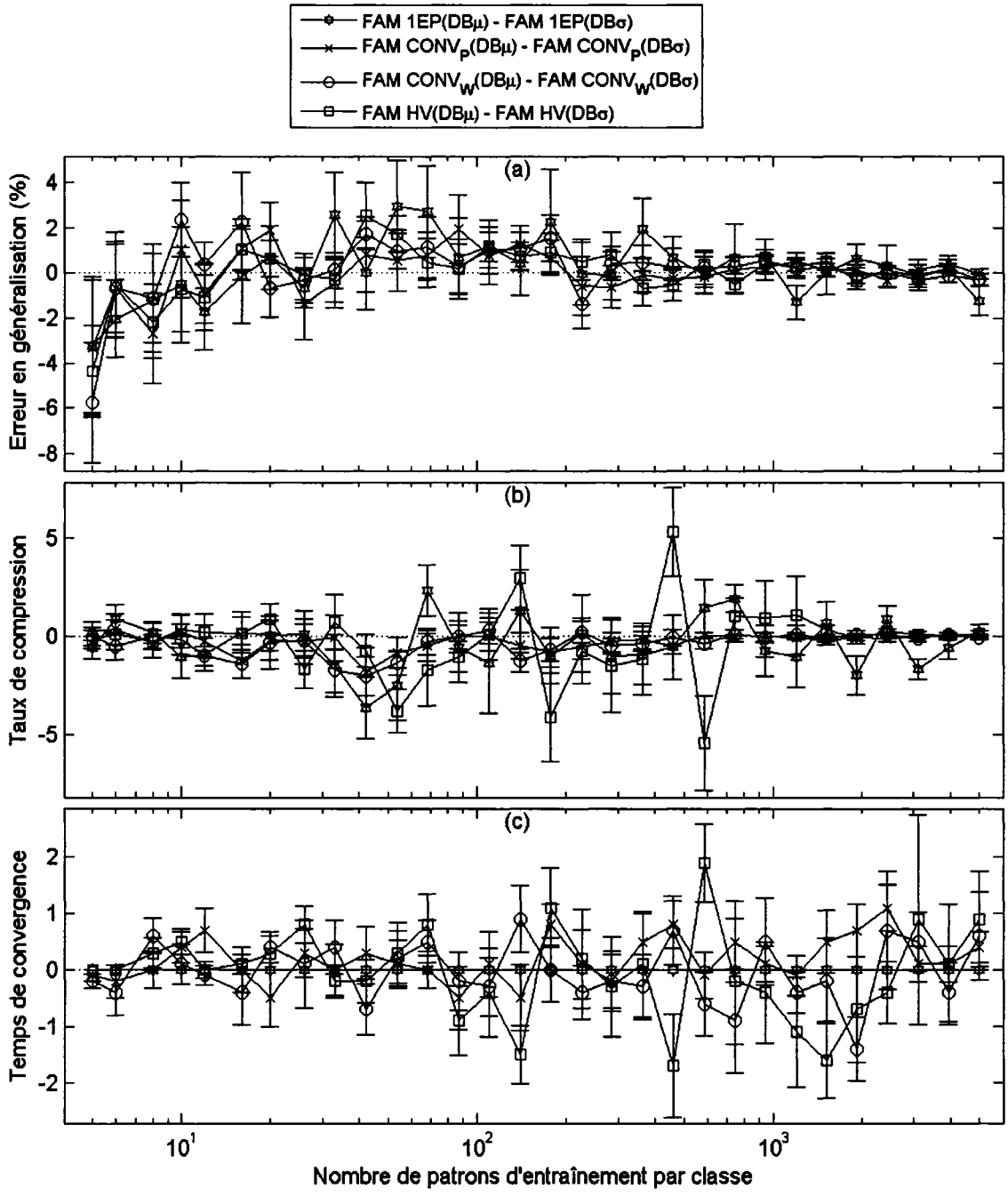


Figure 96 Différence des performances du FAM entre DBμ(15%) et DB$_\sigma$(15%) (a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

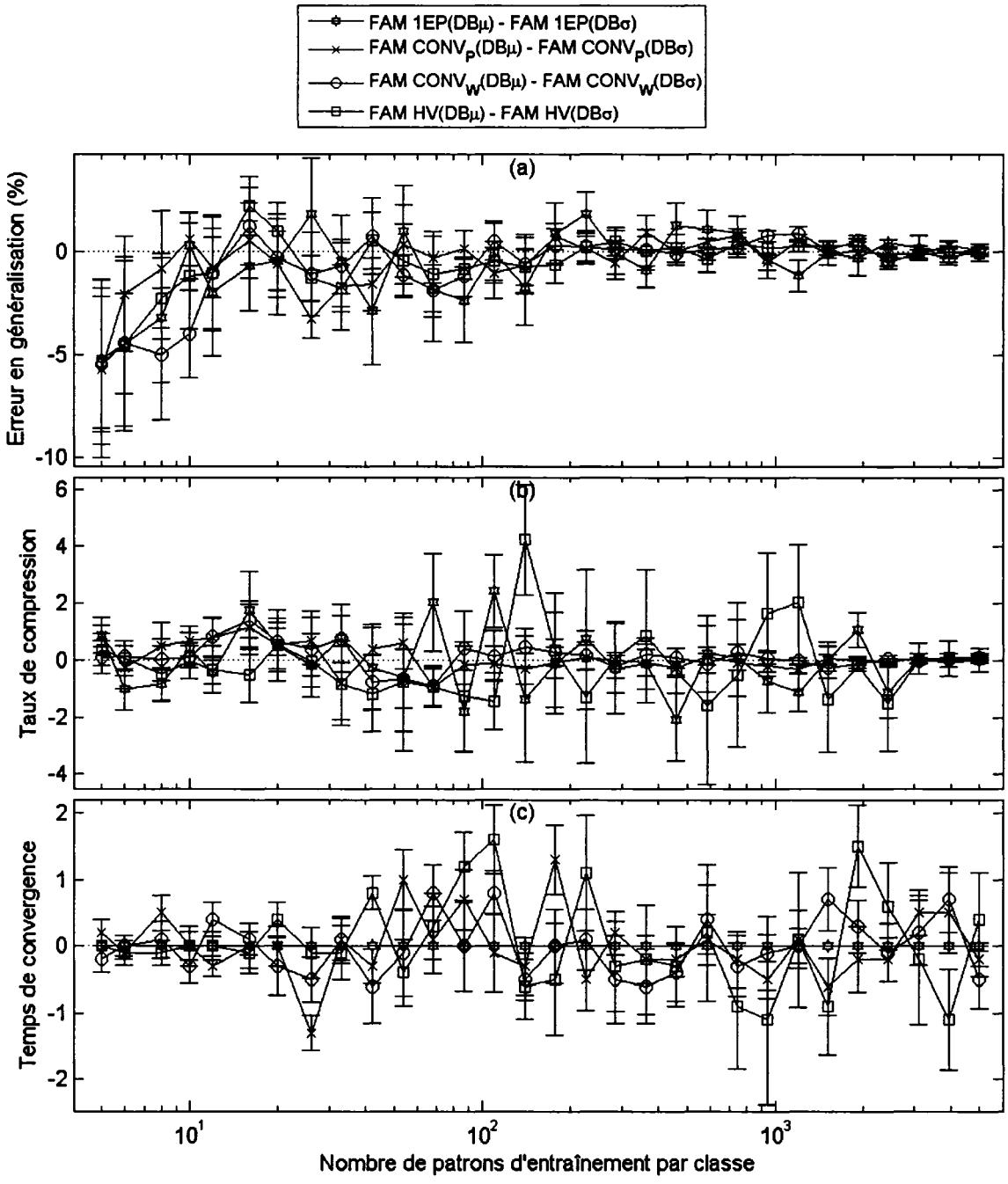


Figure 97 Différence des performances du FAM entre DBμ(17%) et DB$_\sigma$(17%)
(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

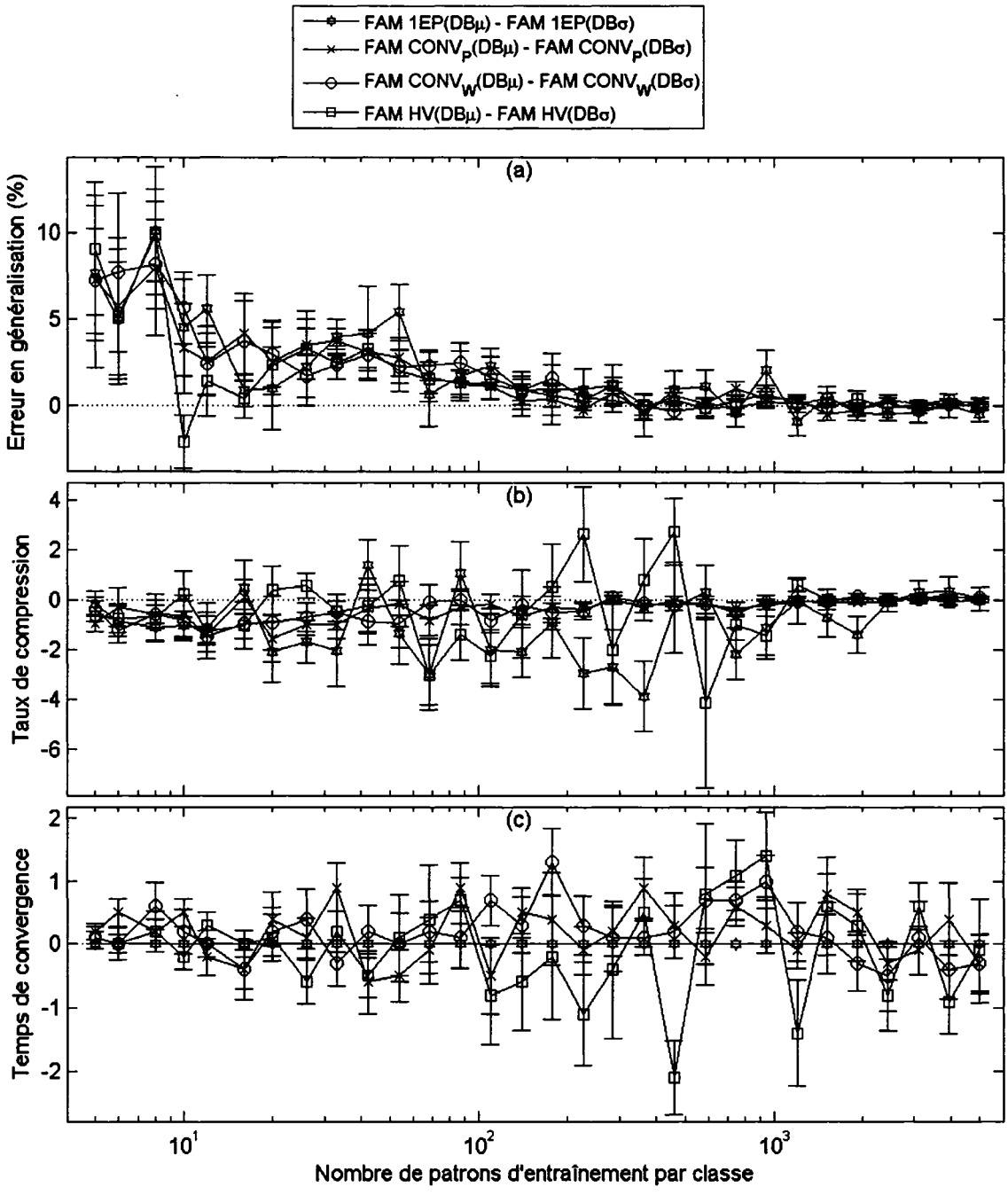


Figure 98 Différence des performances du FAM entre DBμ(19%) et DB$_\sigma$(19%) (a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

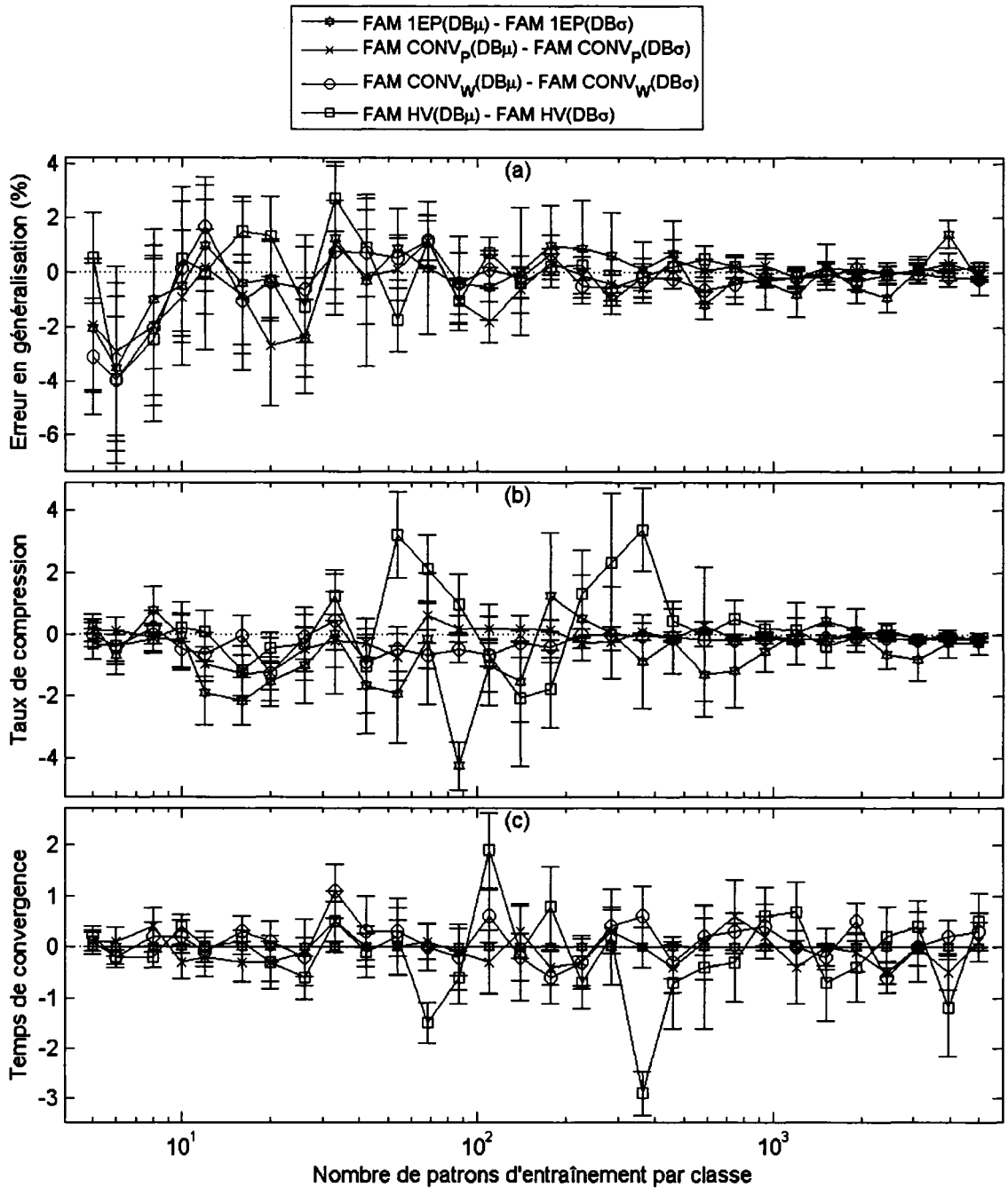


Figure 99 Différence des performances du FAM entre DBμ(21%) et DB$_\sigma$(21%)
(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

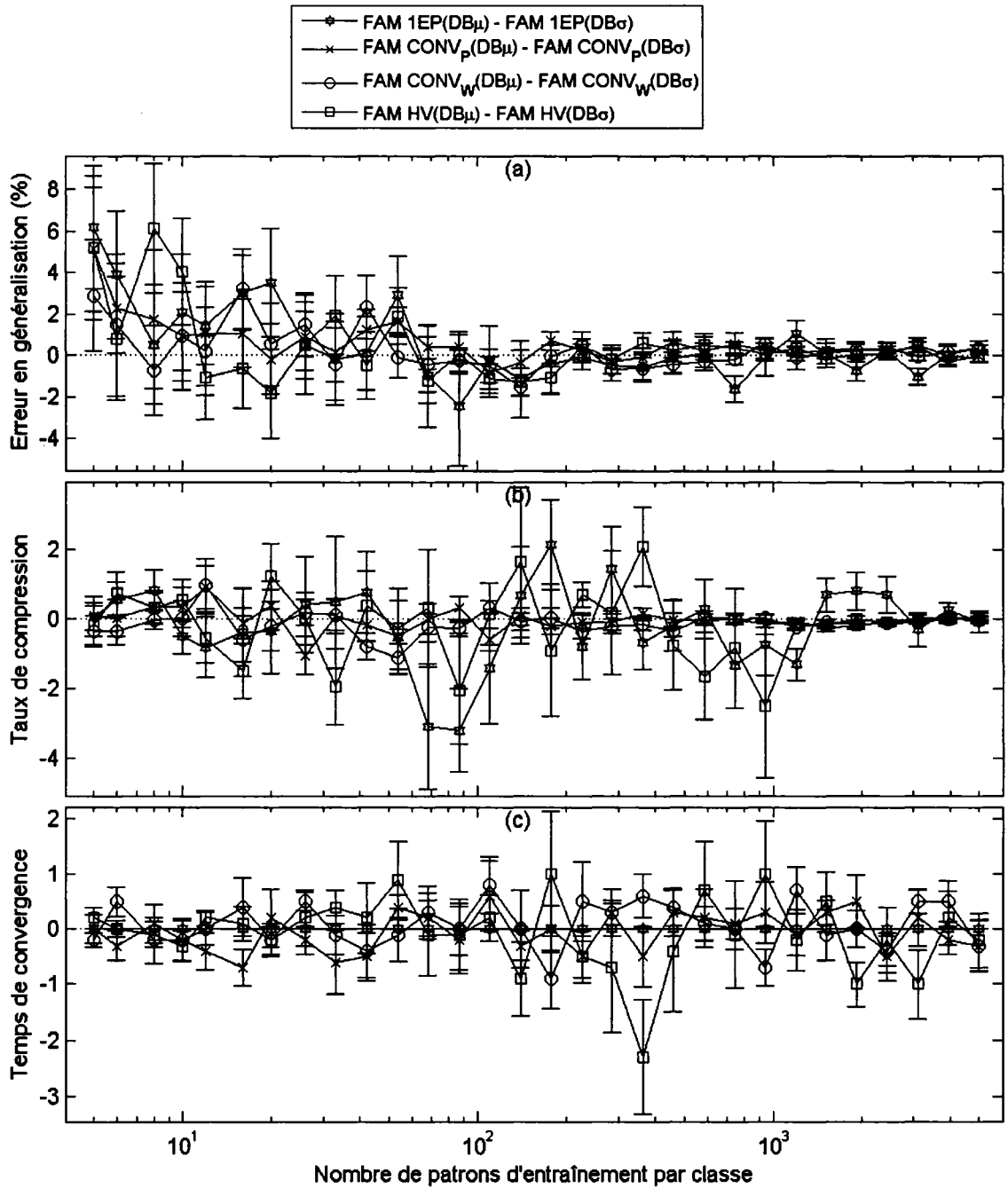


Figure 100 Différence des performances du FAM entre DBμ(23%) et DB$_\sigma$(23%) (a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

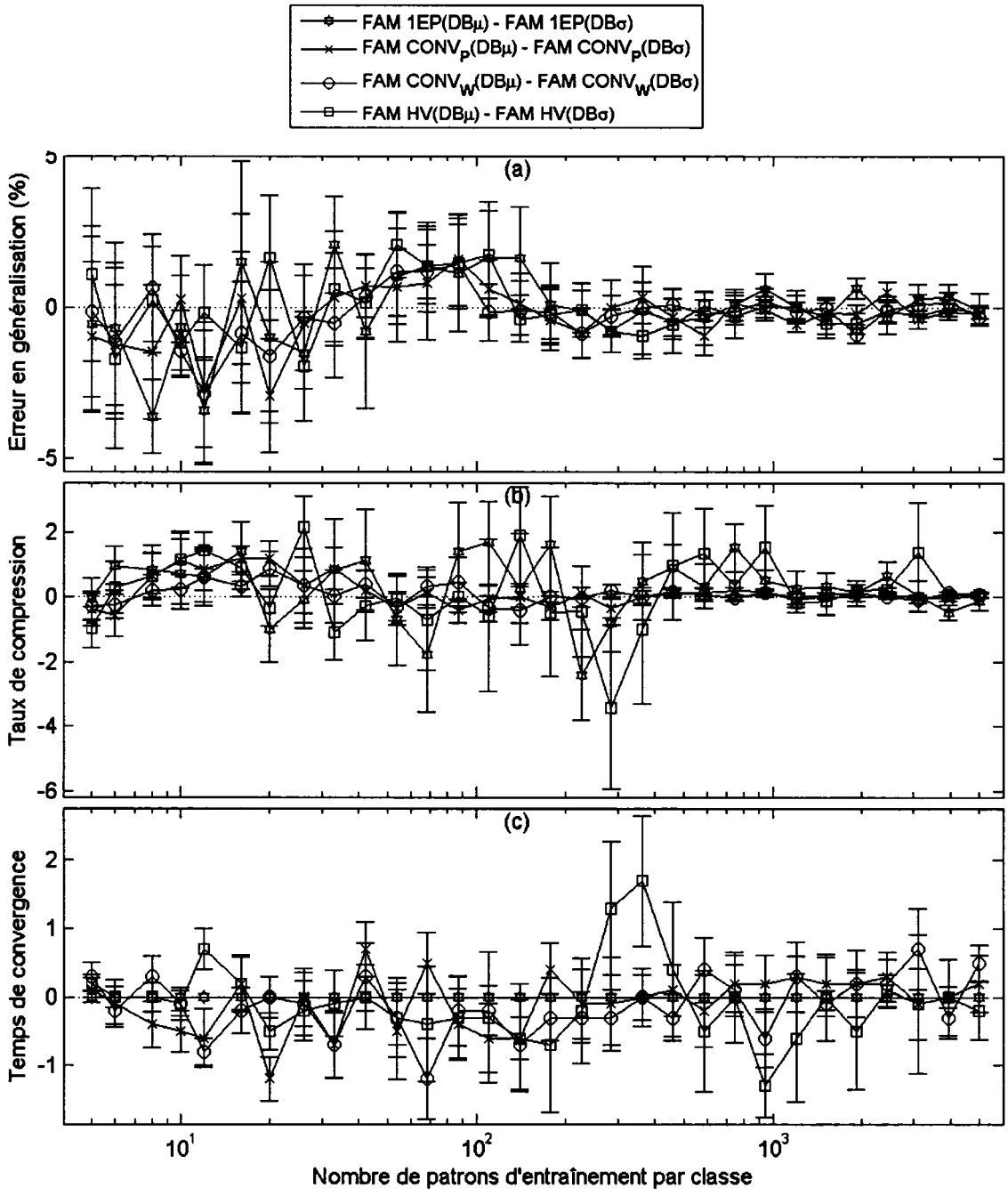


Figure 101 Différence des performances du FAM entre DBμ(25%) et DB$_\sigma$(25%) (a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

ANNEXE 5

Effets de la technique de normalisation

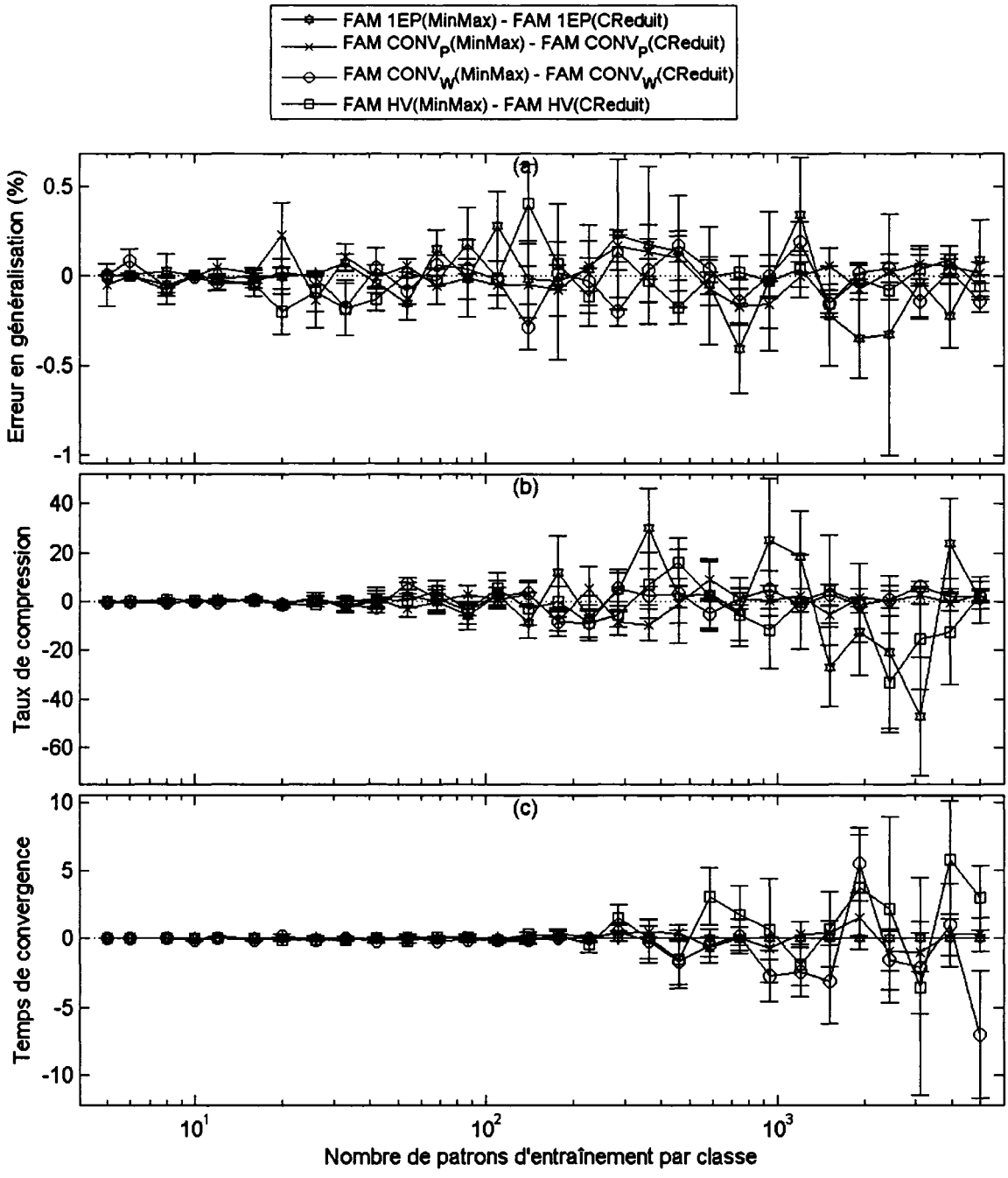


Figure 102 Différence des performances du FAM entre MinMax et CReduite avec la base DBμ(1%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

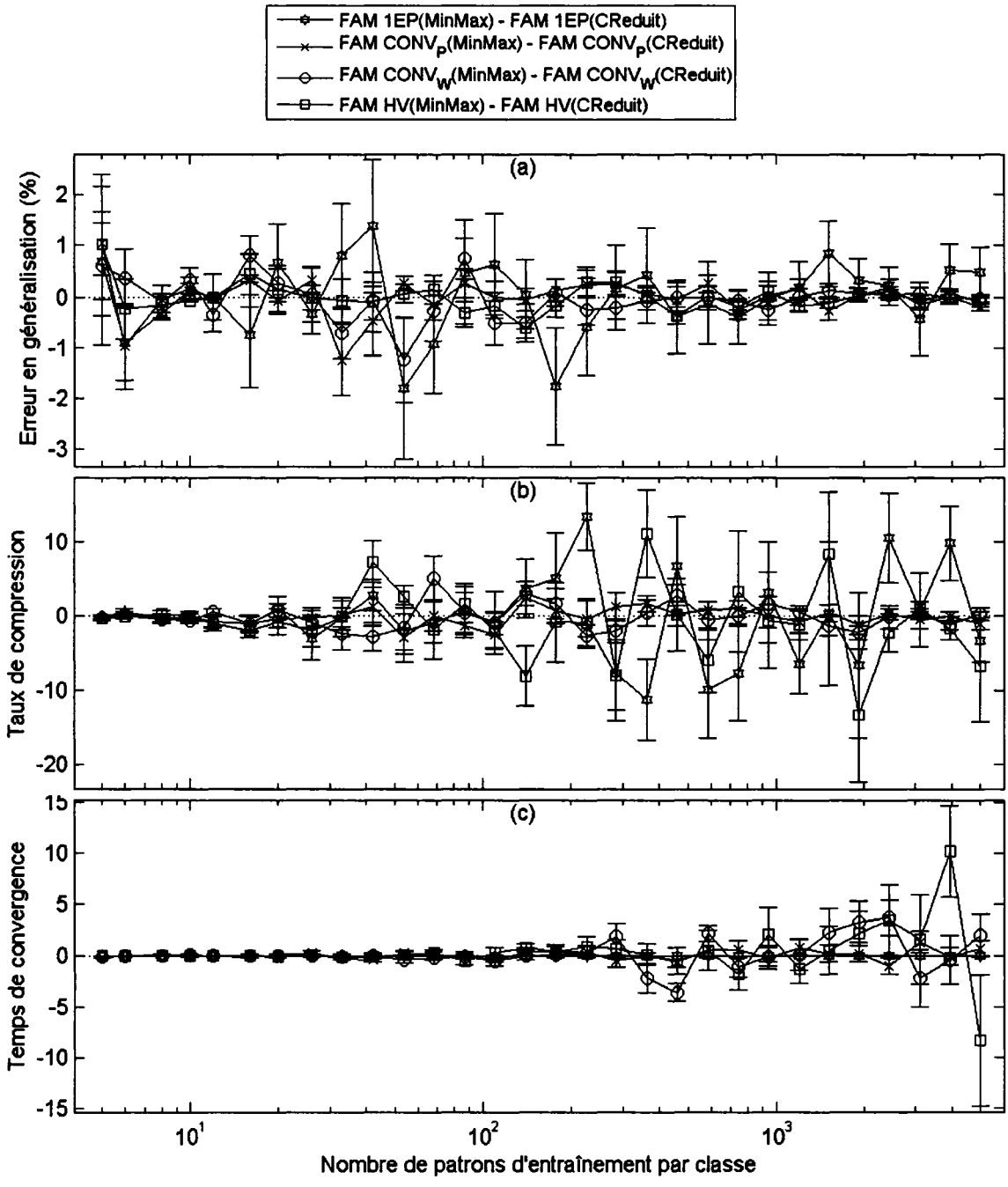


Figure 103 Différence des performances du FAM entre MinMax et CReduite avec la base DBμ(3%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

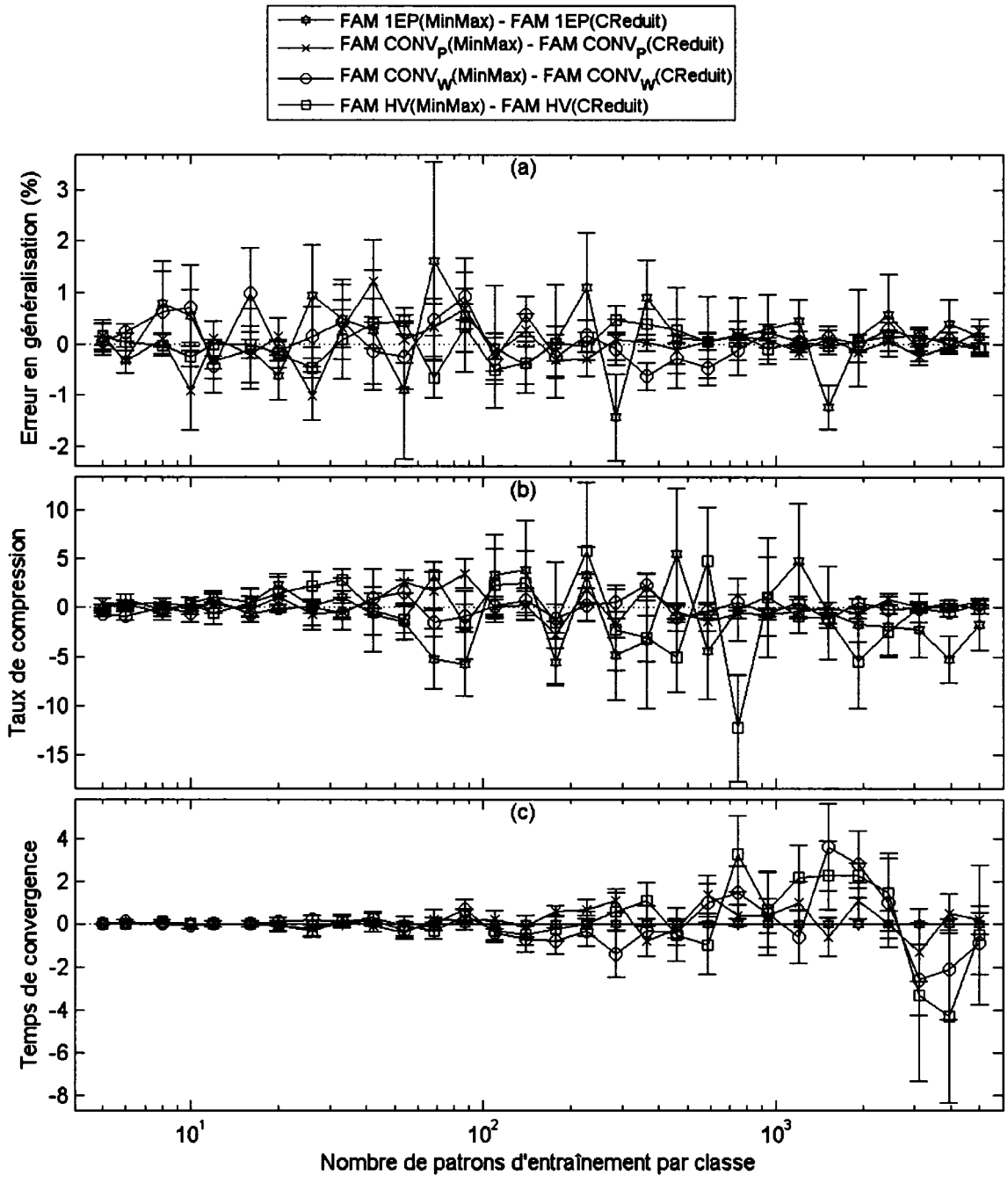


Figure 104 Différence des performances du FAM entre MinMax et CReduite avec la base DBμ(5%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

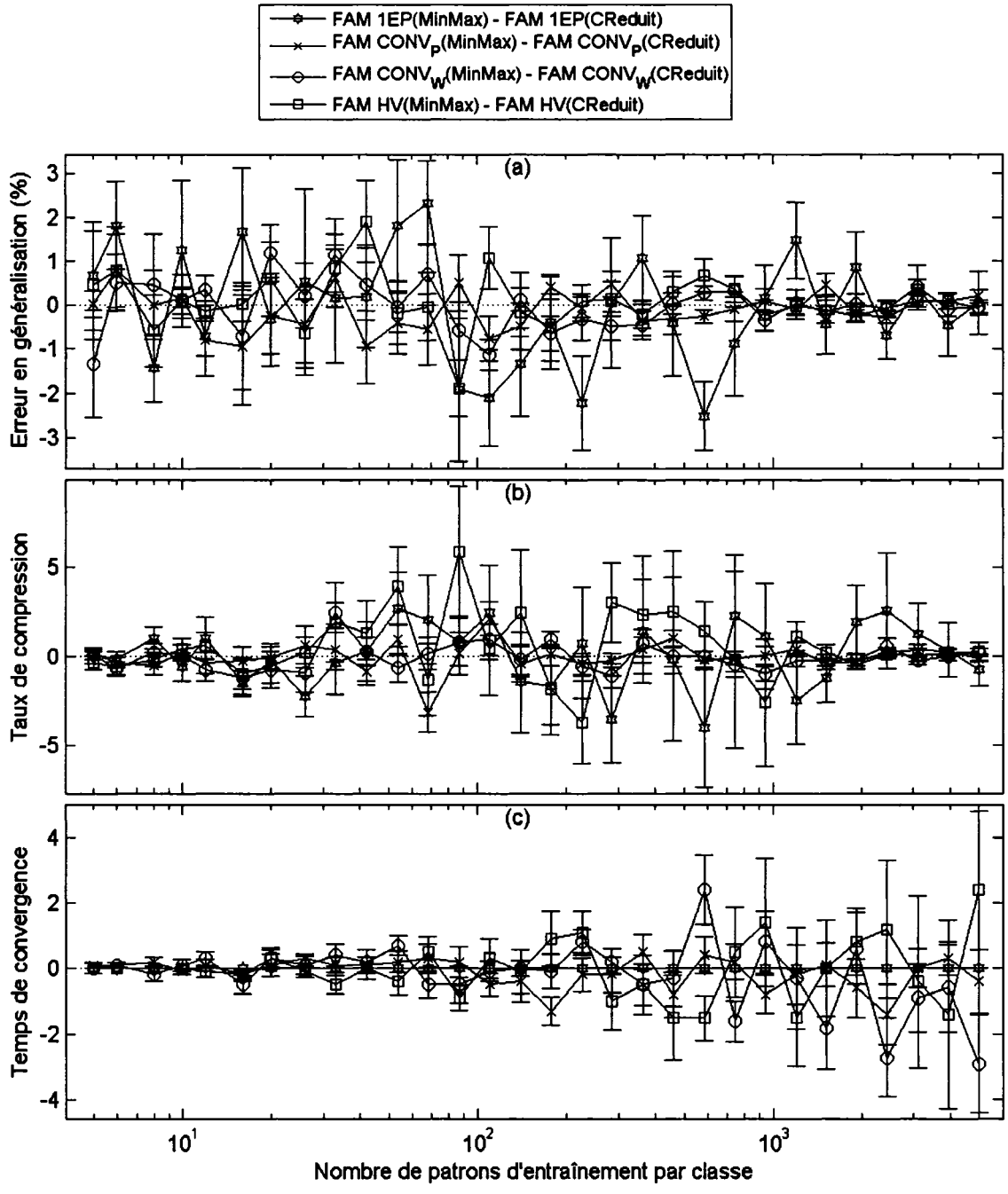


Figure 105 Différence des performances du FAM entre MinMax et CReduite avec la base DBμ(7%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

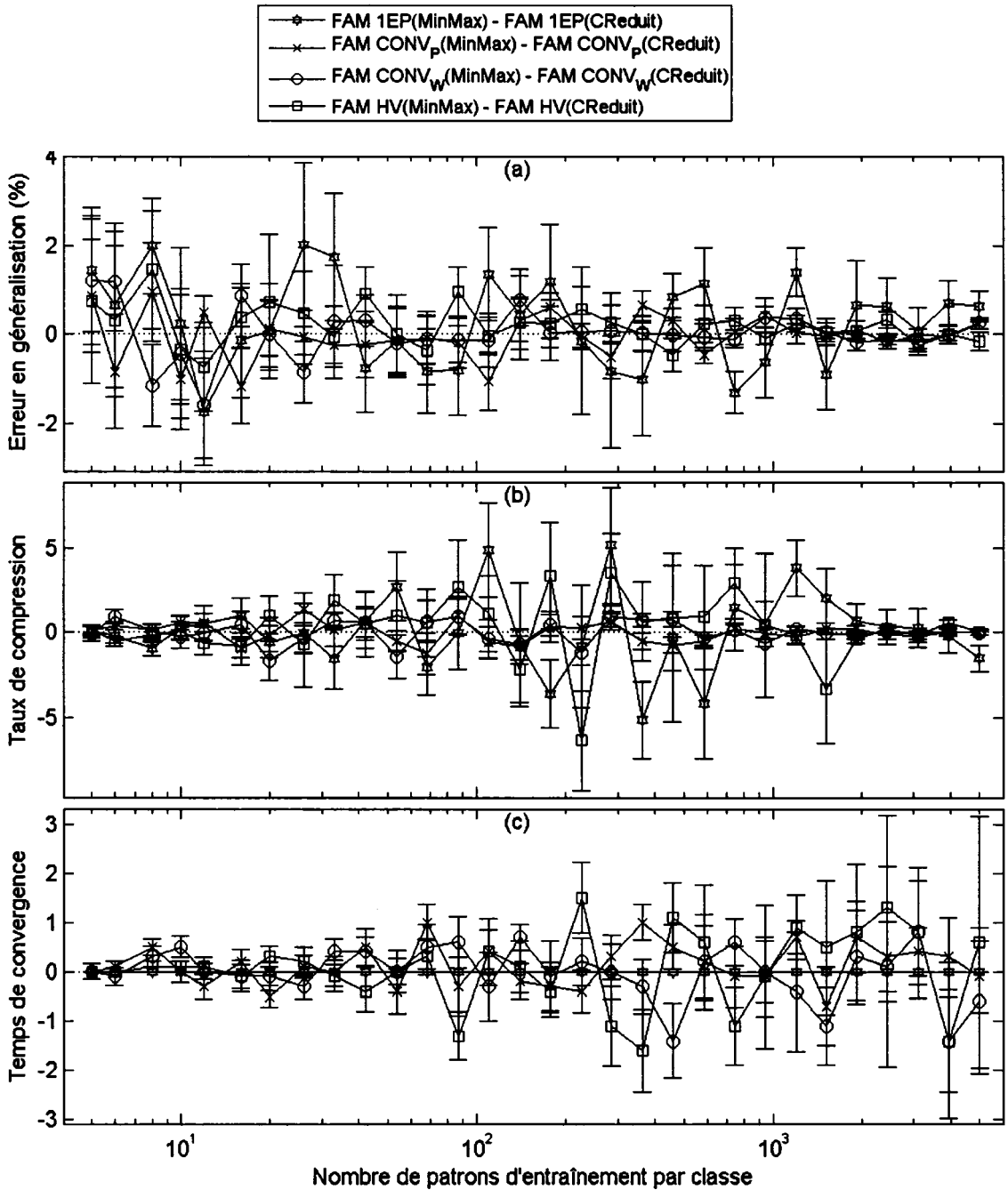


Figure 106 Différence des performances du FAM entre MinMax et CReduite avec la base DBμ(9%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

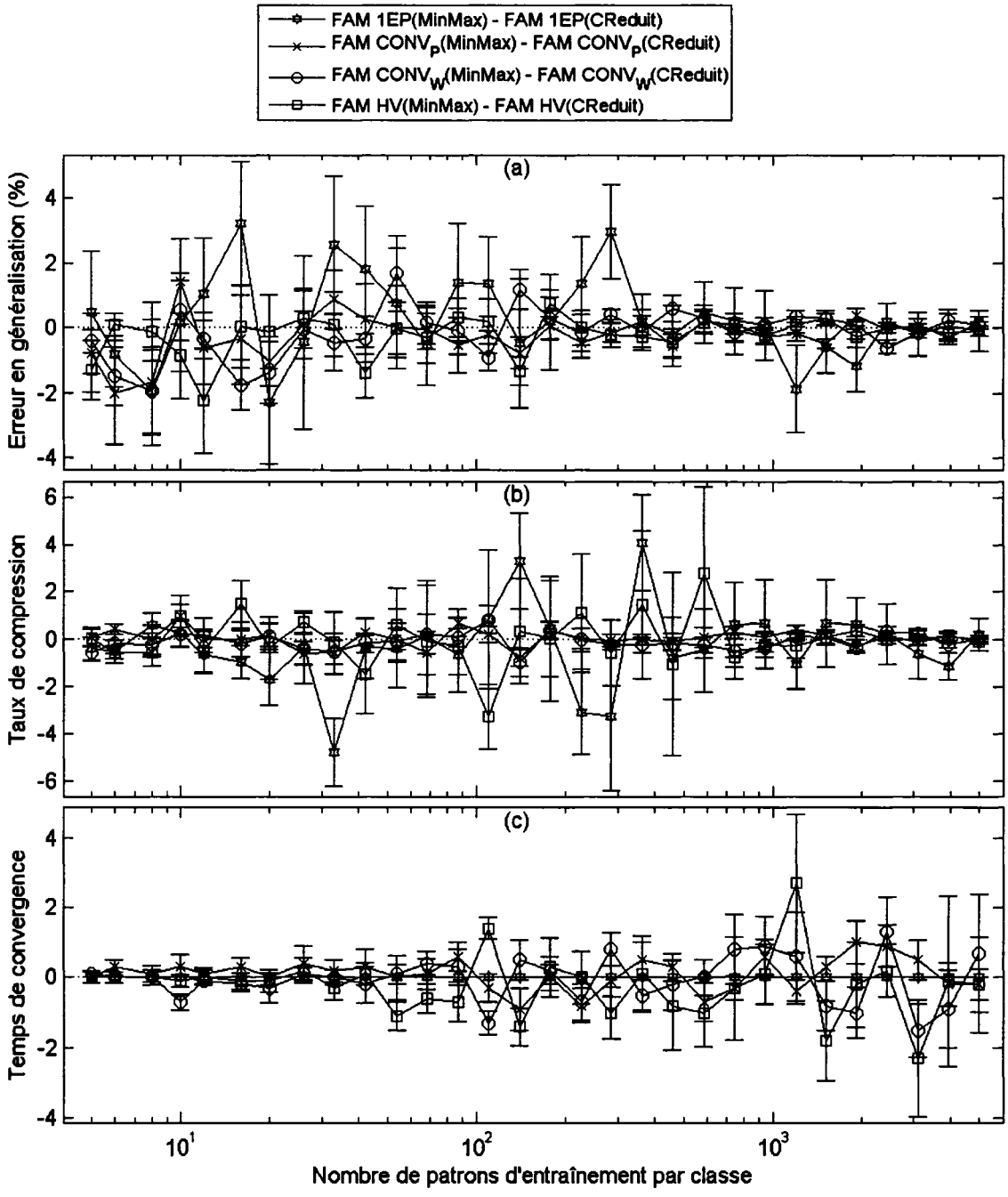


Figure 107 Différence des performances du FAM entre MinMax et CReduite avec la base DBμ(11%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

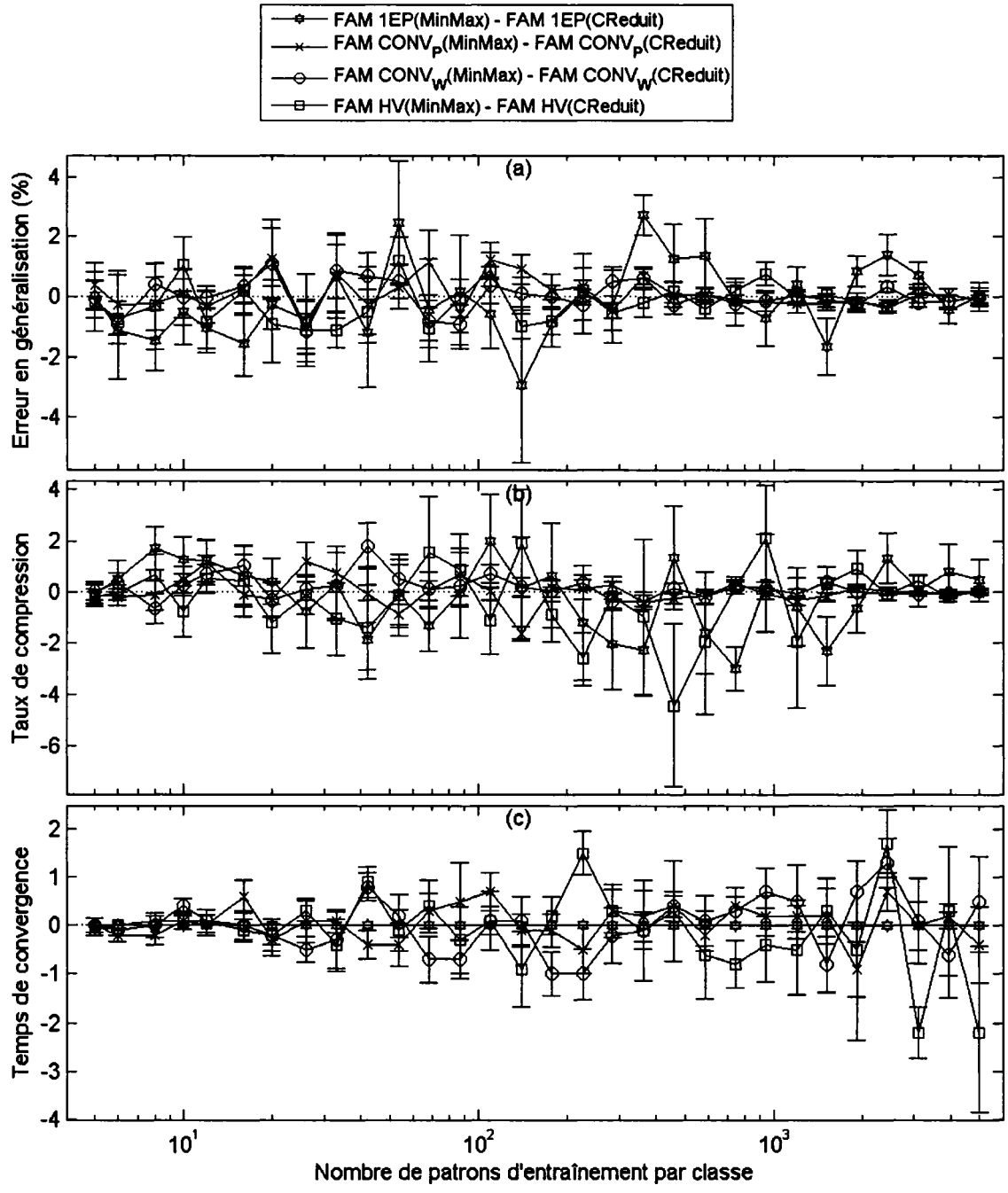


Figure 108 Différence des performances du FAM entre MinMax et CReduite avec la base DBμ(13%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

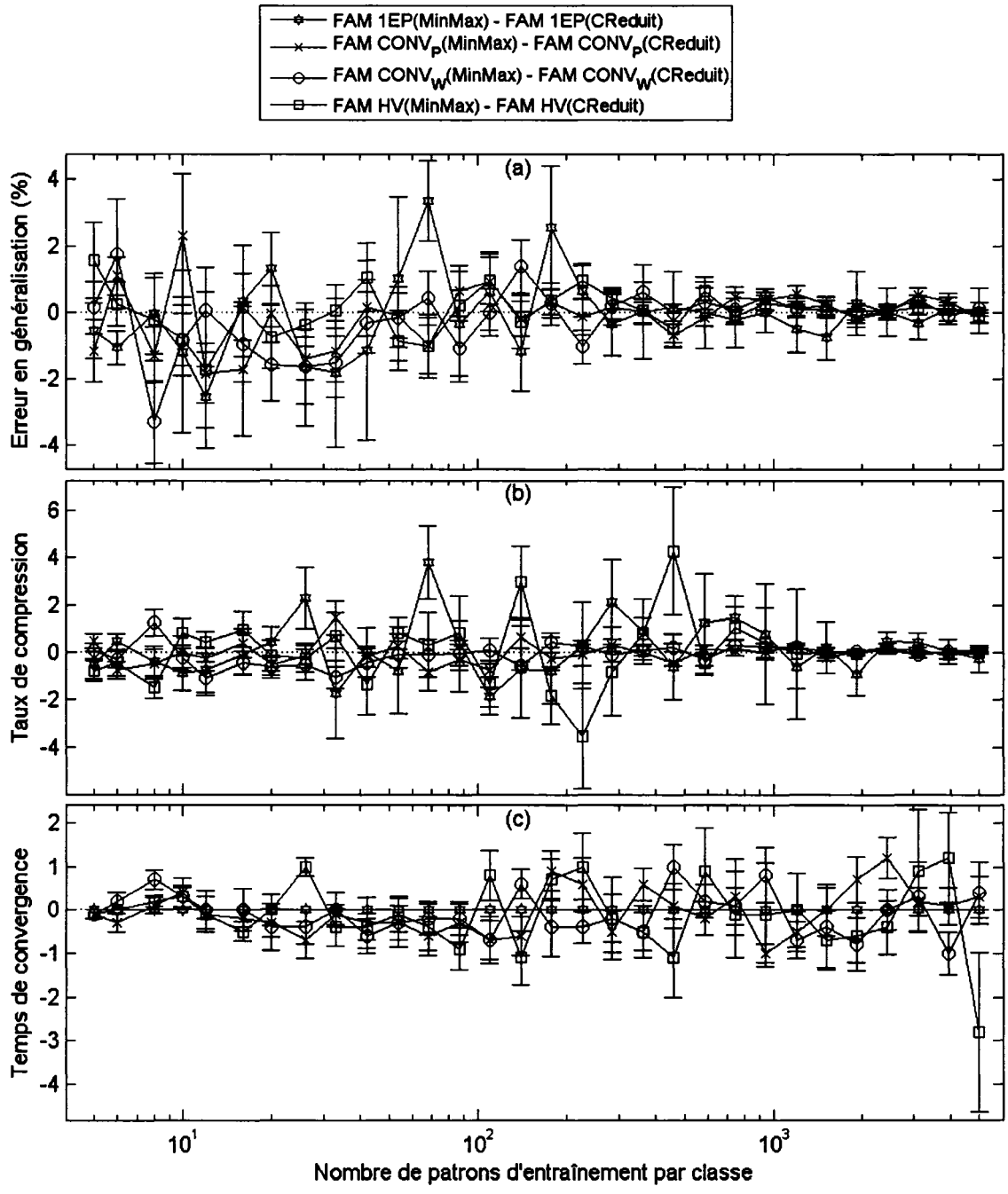


Figure 109 Différence des performances du FAM entre MinMax et CReduite avec la base DBμ(15%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

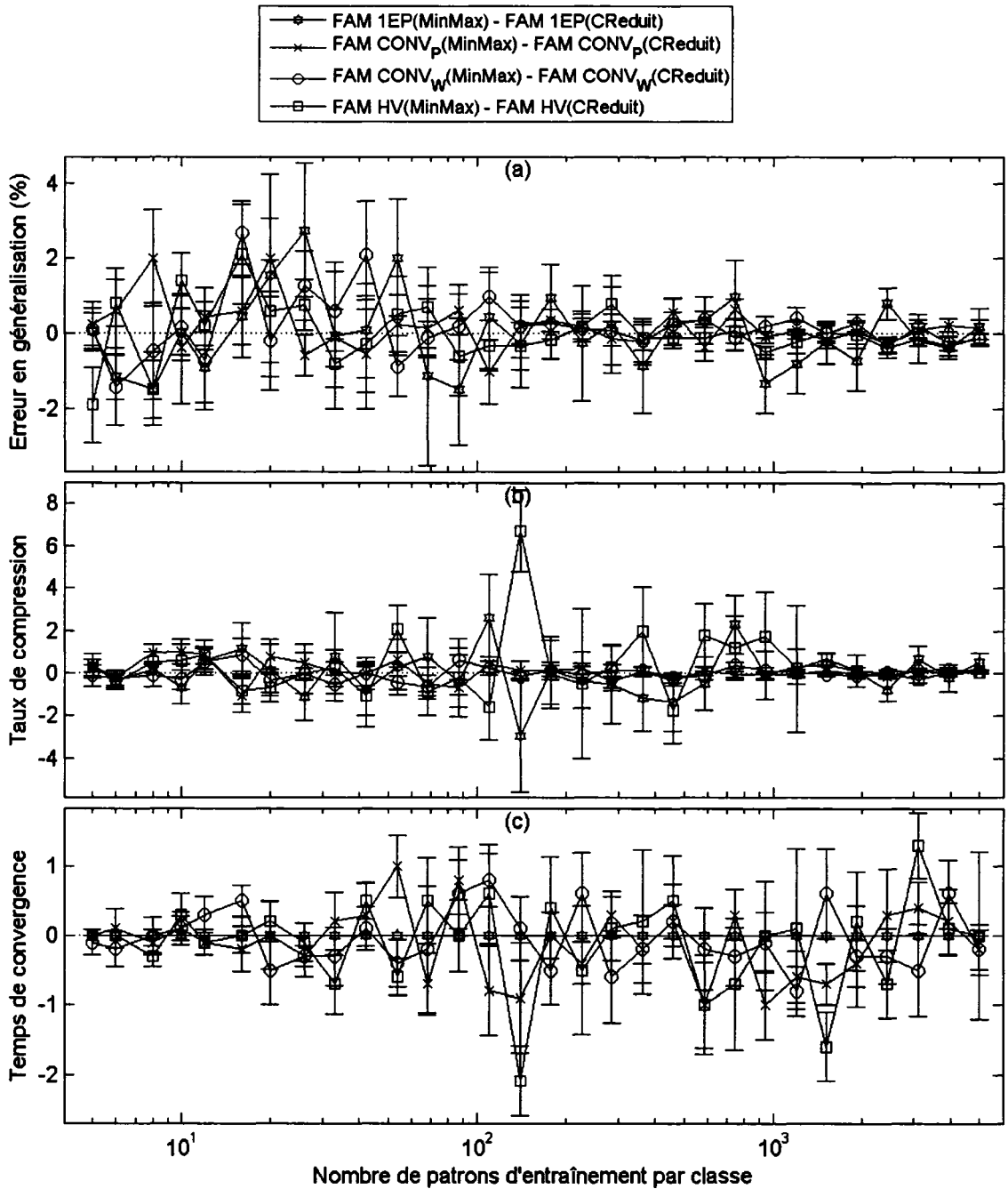


Figure 110 Différence des performances du FAM entre MinMax et CReduite avec la base DBμ(17%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

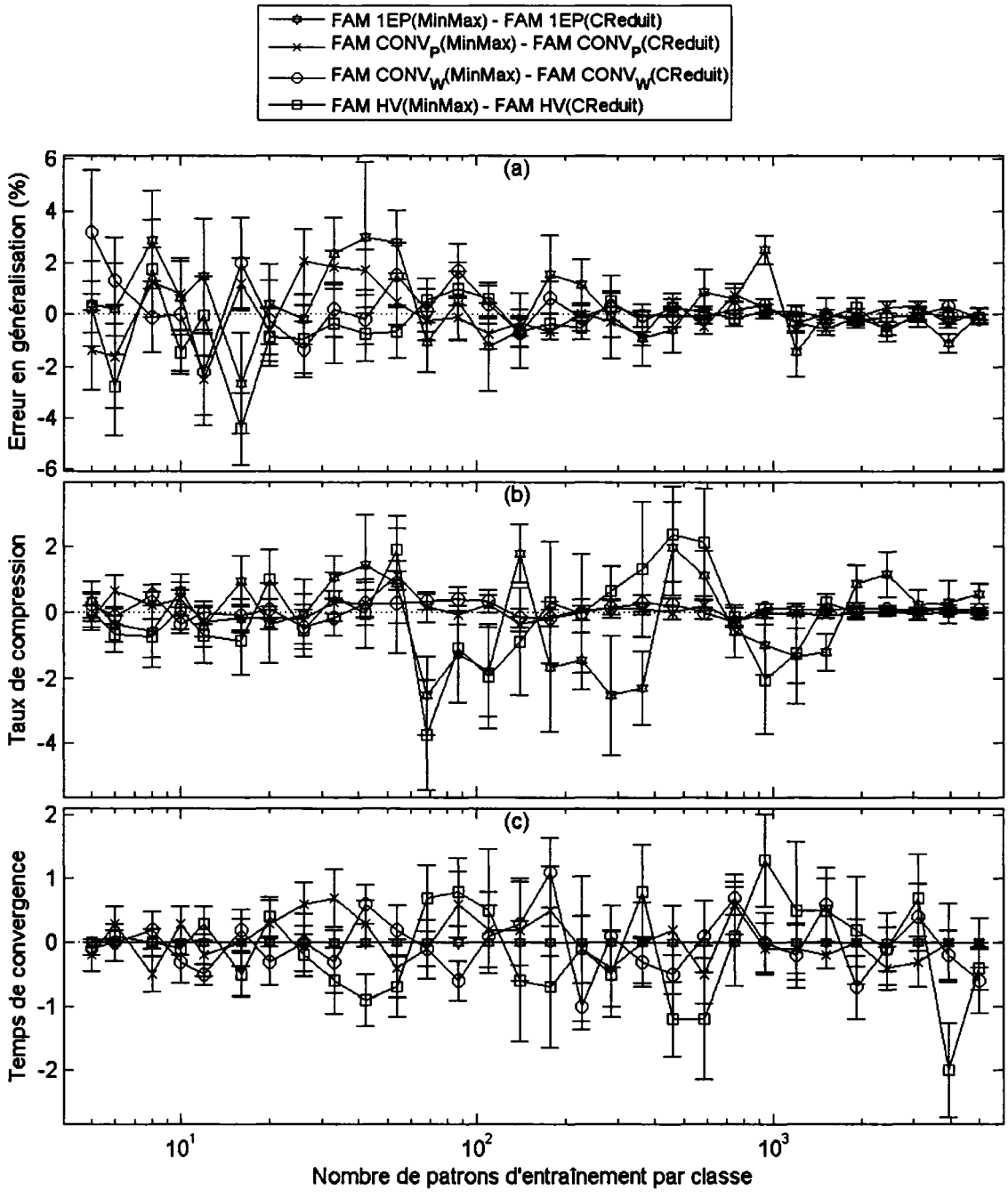


Figure 111 Différence des performances du FAM entre MinMax et CReduite avec la base DBμ(19%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

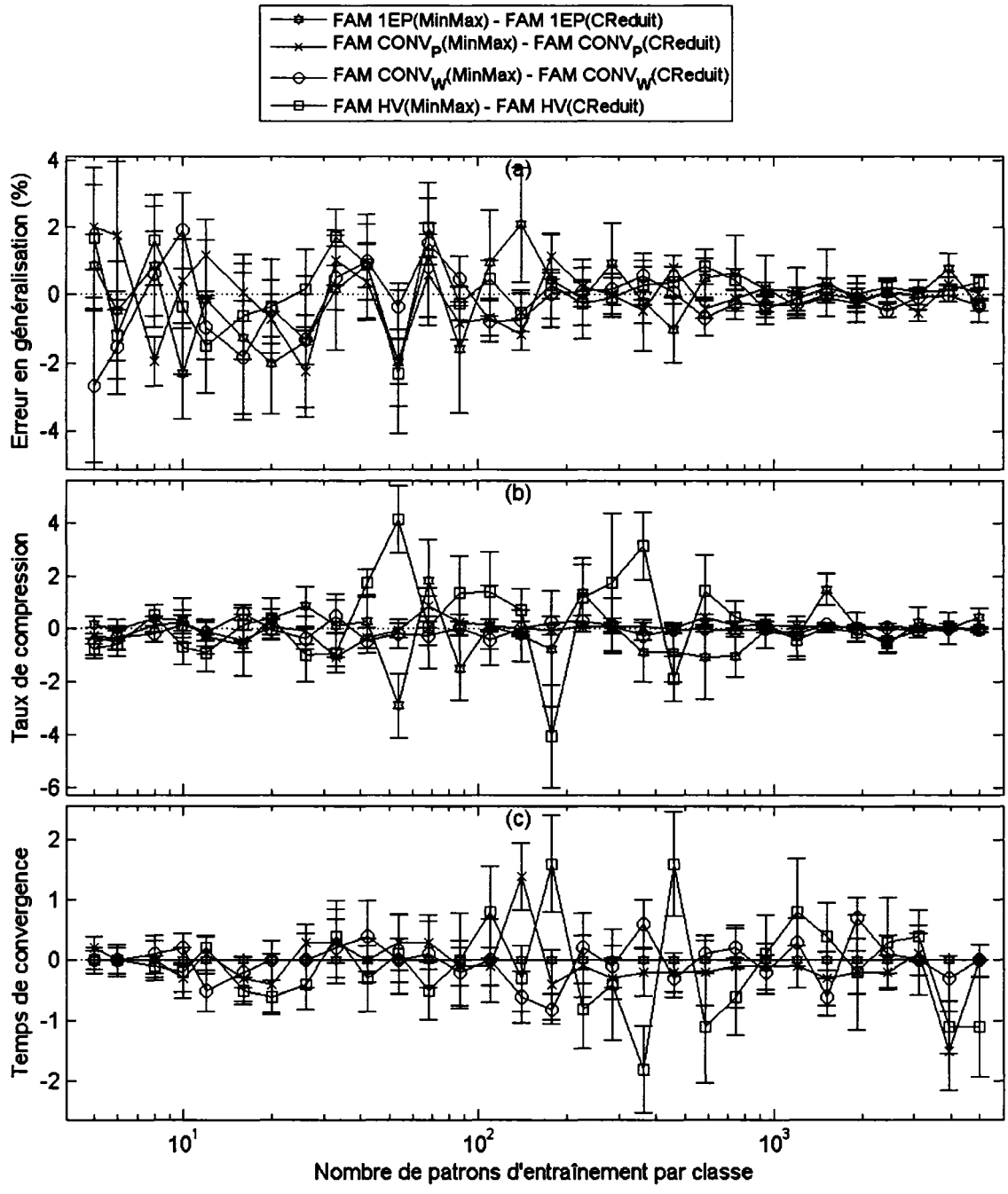


Figure 112 Différence des performances du FAM entre MinMax et CReduite avec la base DBμ(21%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

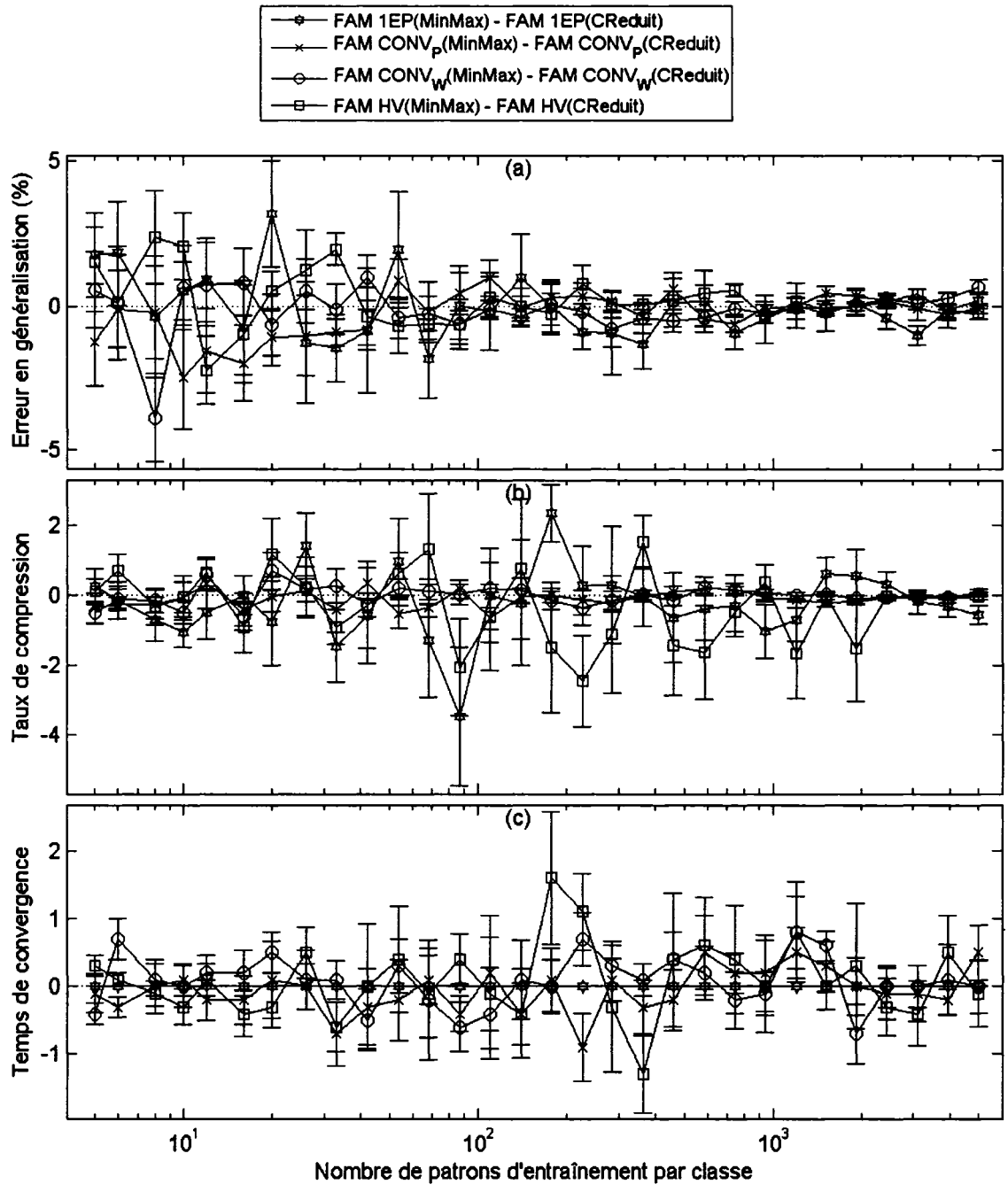


Figure 113 Différence des performances du FAM entre MinMax et CReduite avec la base DBμ(23%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

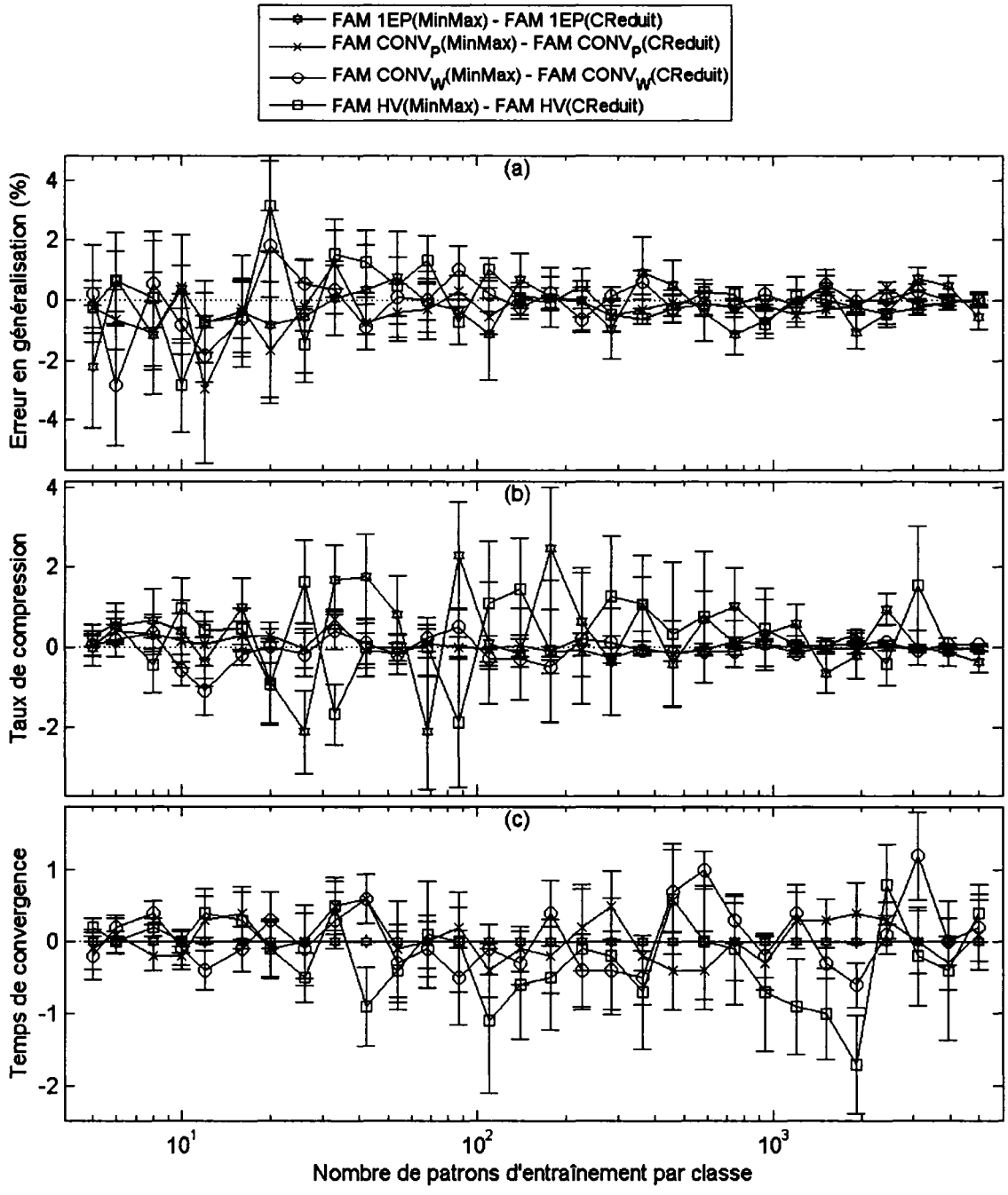


Figure 114 Différence des performances du FAM entre MinMax et CReduite avec la base DBμ(25%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

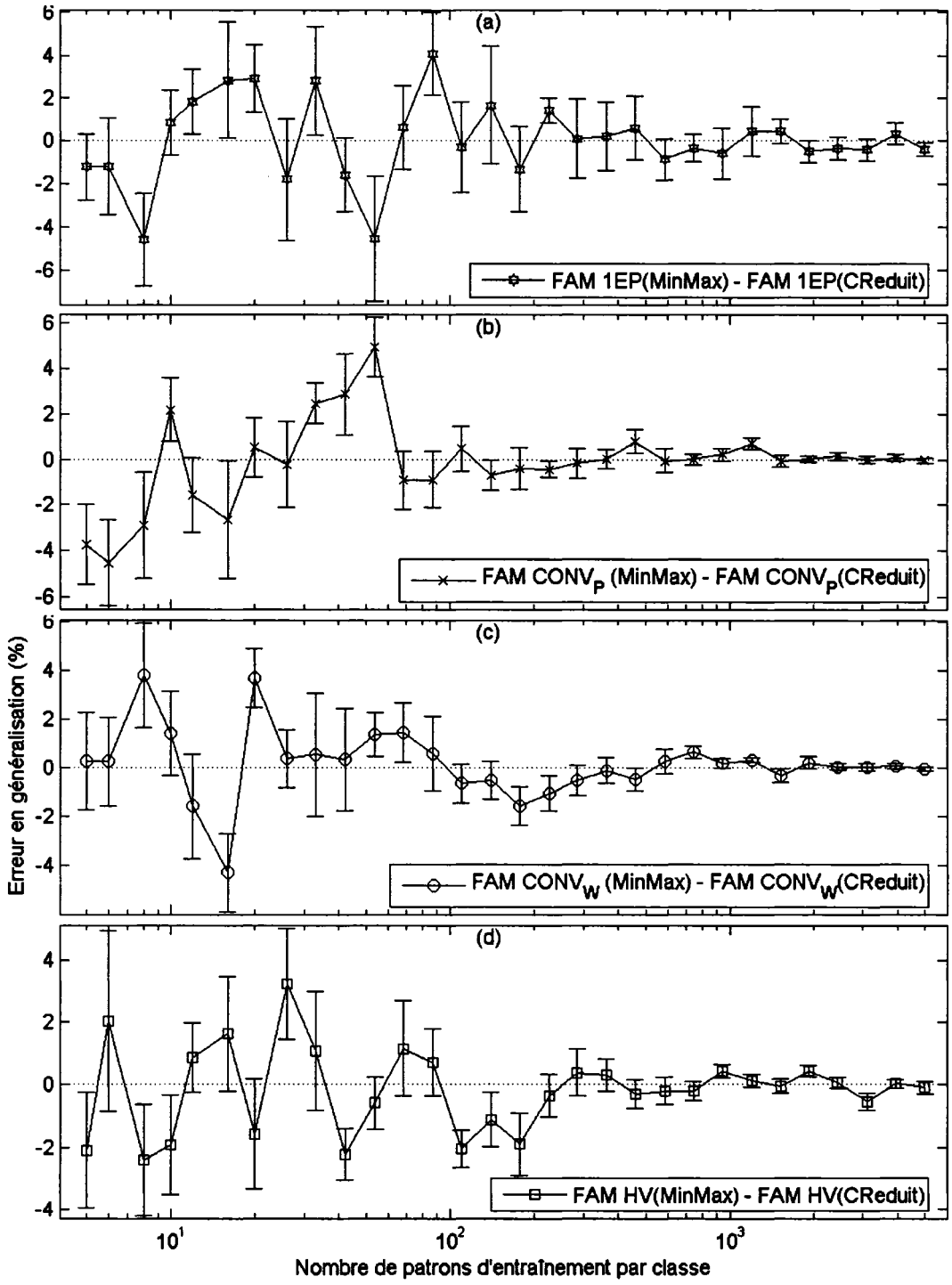


Figure 115 Différence entre l'erreur en généralisation avec la base DB_{P2}

(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

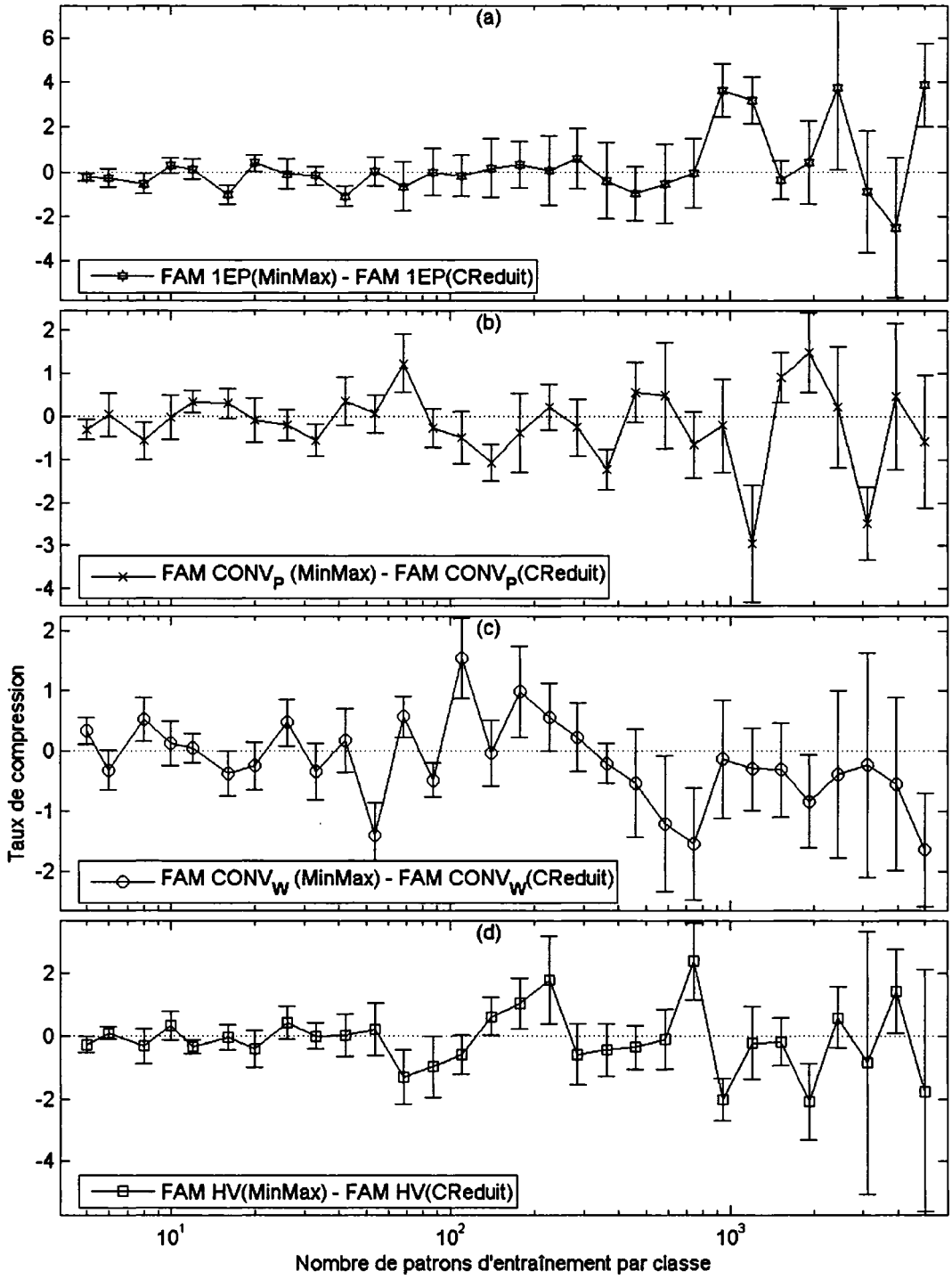


Figure 116 Différence sur le taux de compression avec la base DB_{P2}

(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

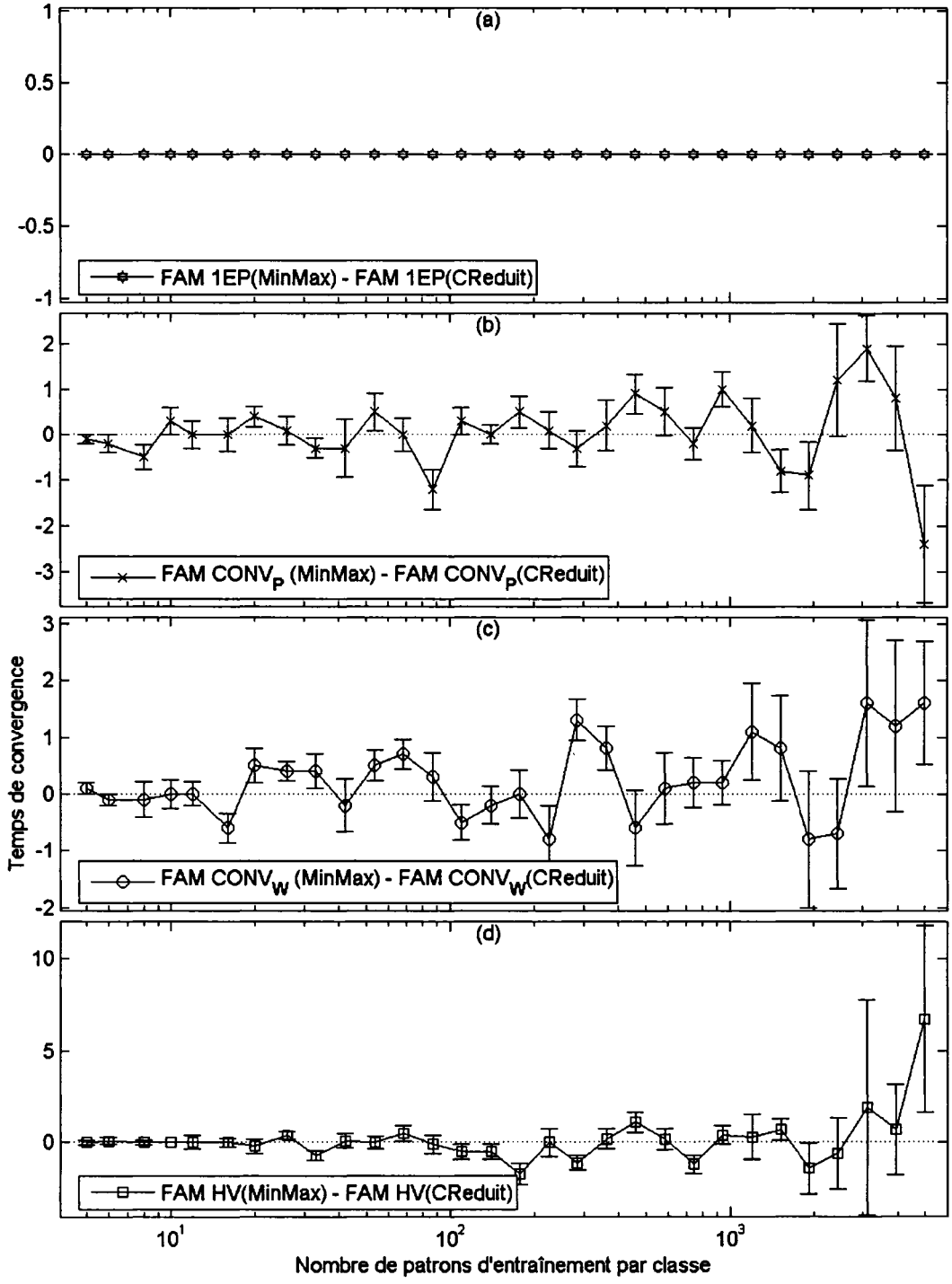


Figure 117 Différence sur le temps de convergence avec la base DB_{P2}

(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

ANNEXE 6

Effets de la polarité du Match Tracking

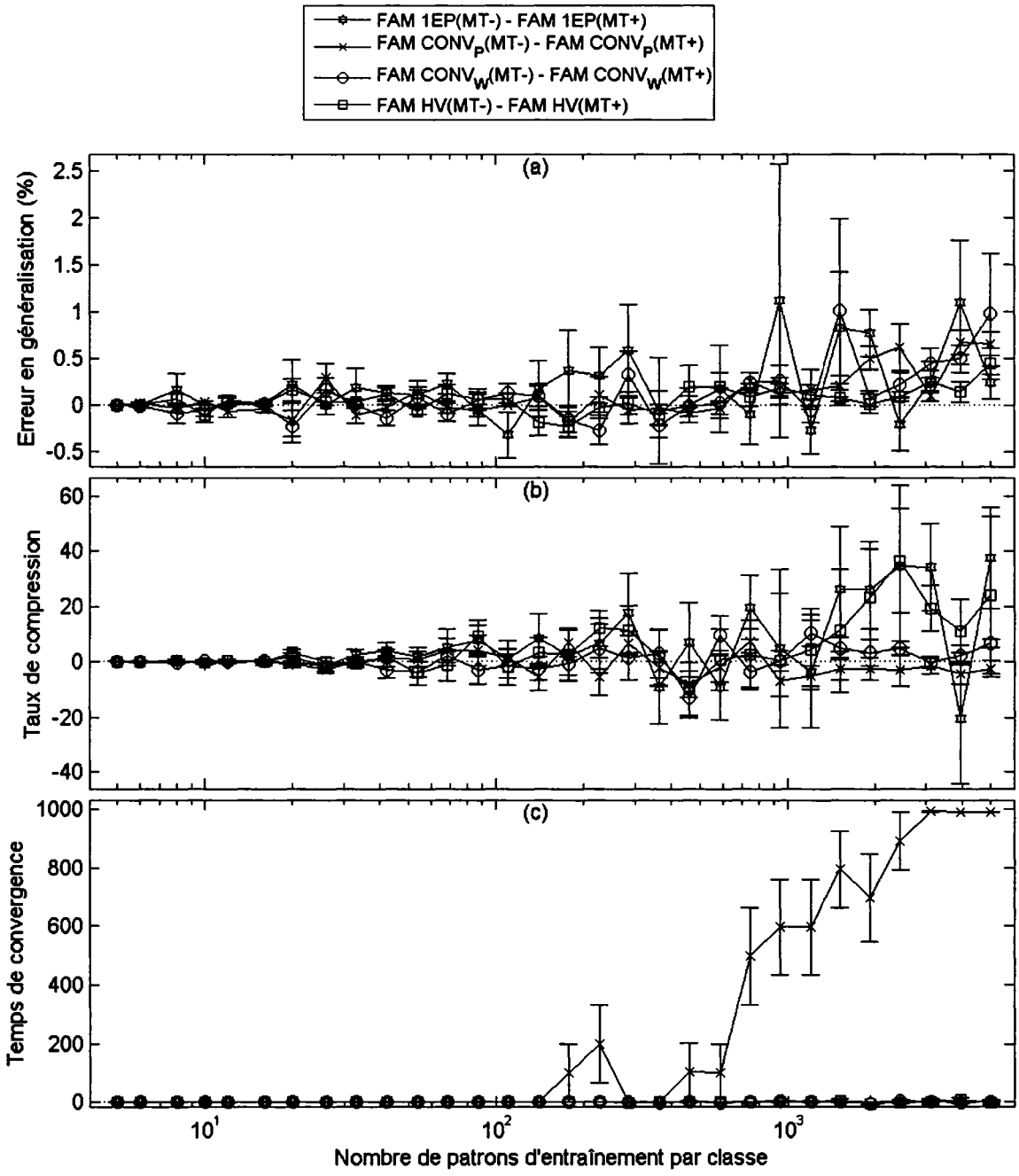


Figure 118 Différence des performances du FAM entre MT- et MT+ avec la base DBμ(1%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

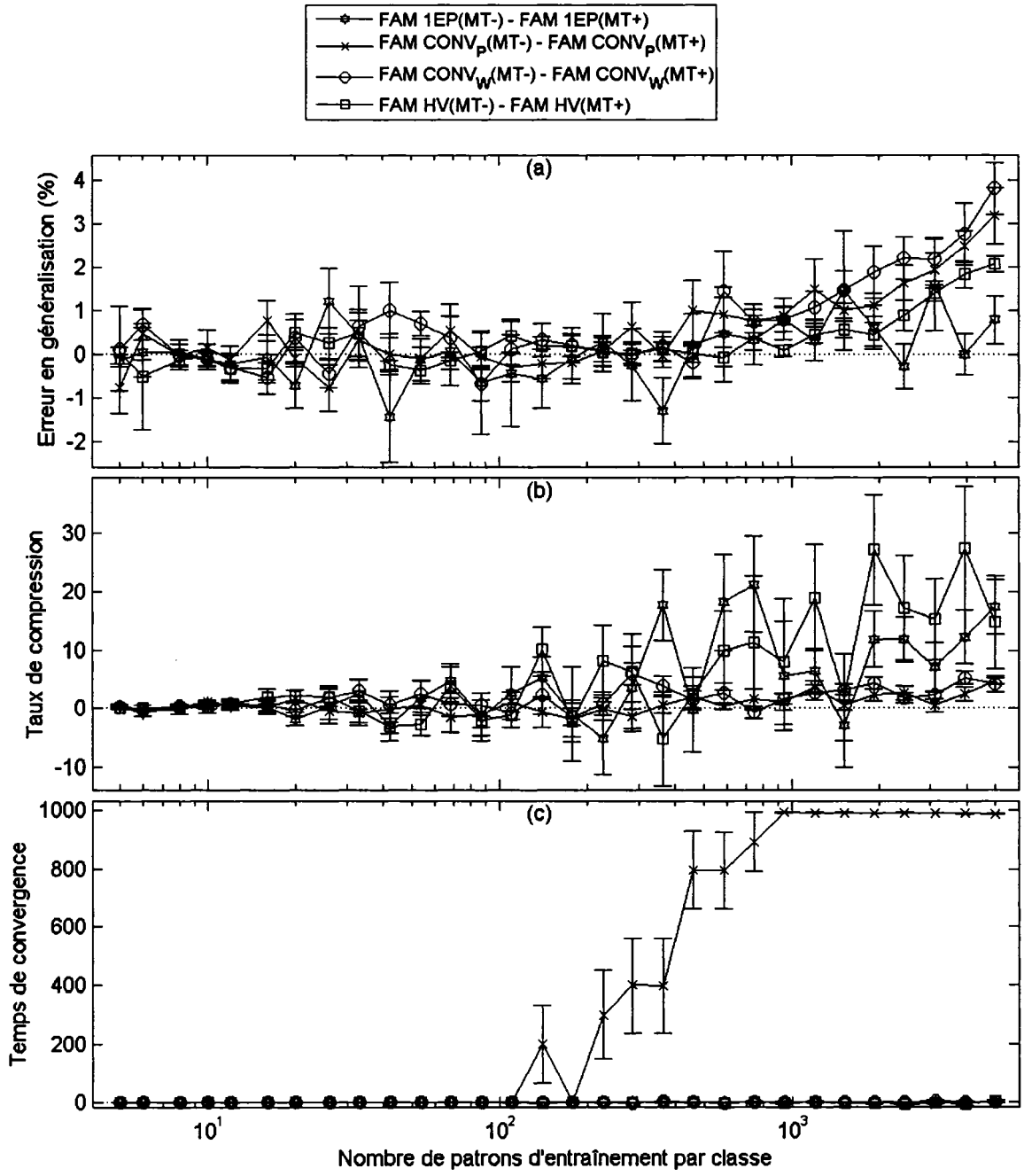


Figure 119 Différence des performances du FAM entre MT- et MT+ avec la base DBμ(3%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

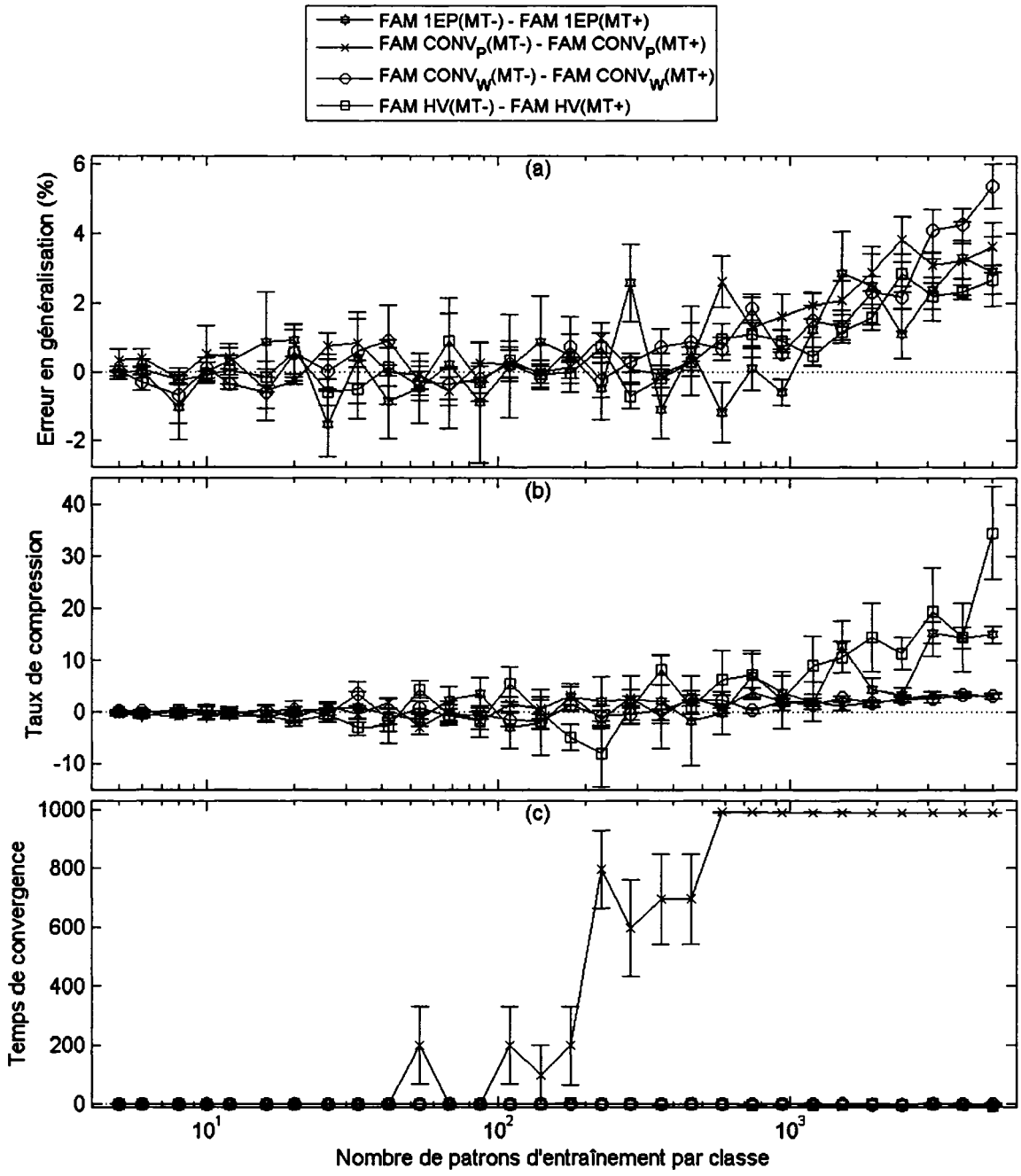


Figure 120 Différence des performances du FAM entre MT- et MT+ avec la base DBμ(5%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

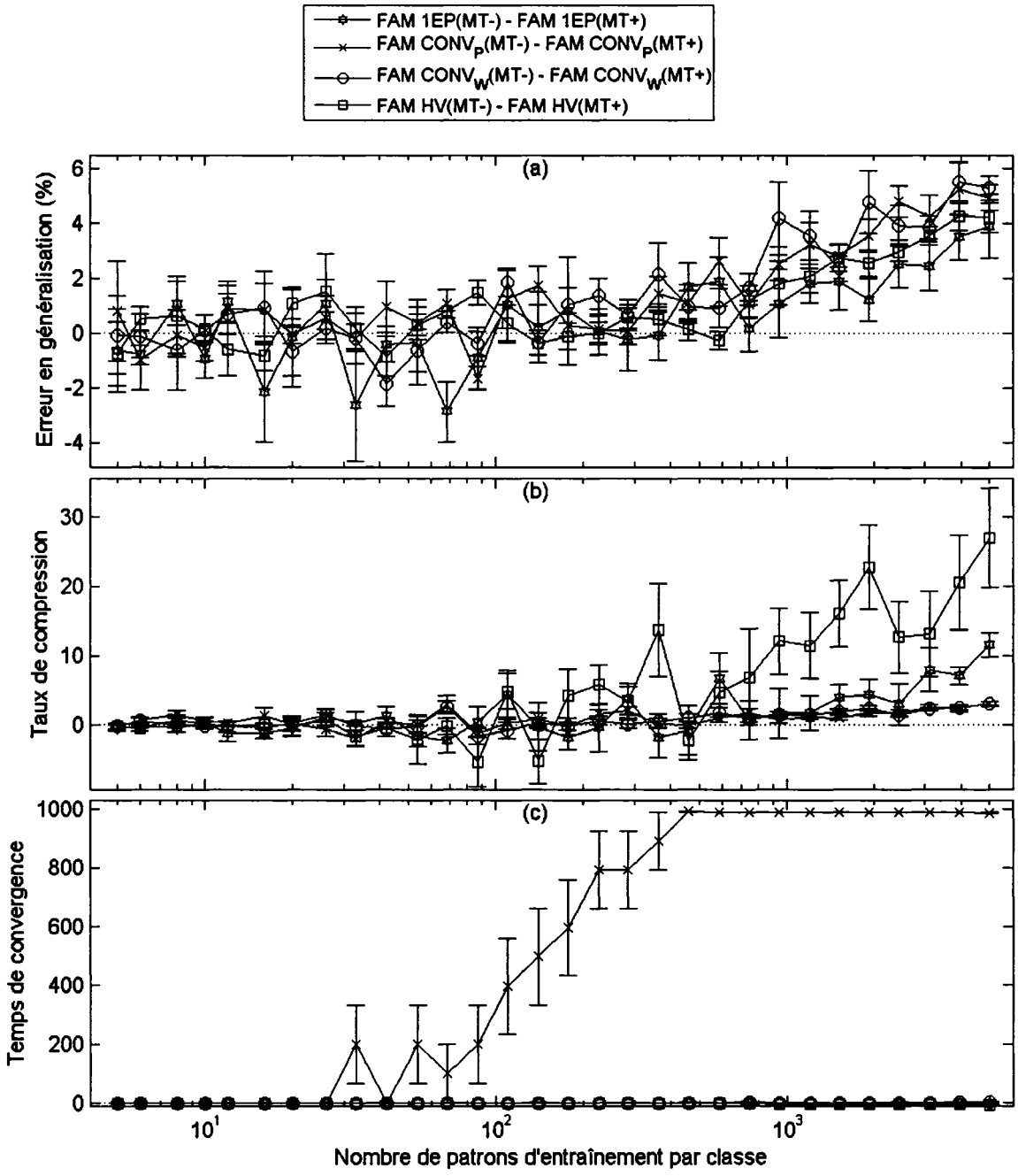


Figure 121 Différence des performances du FAM entre MT- et MT+ avec la base DBμ(7%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

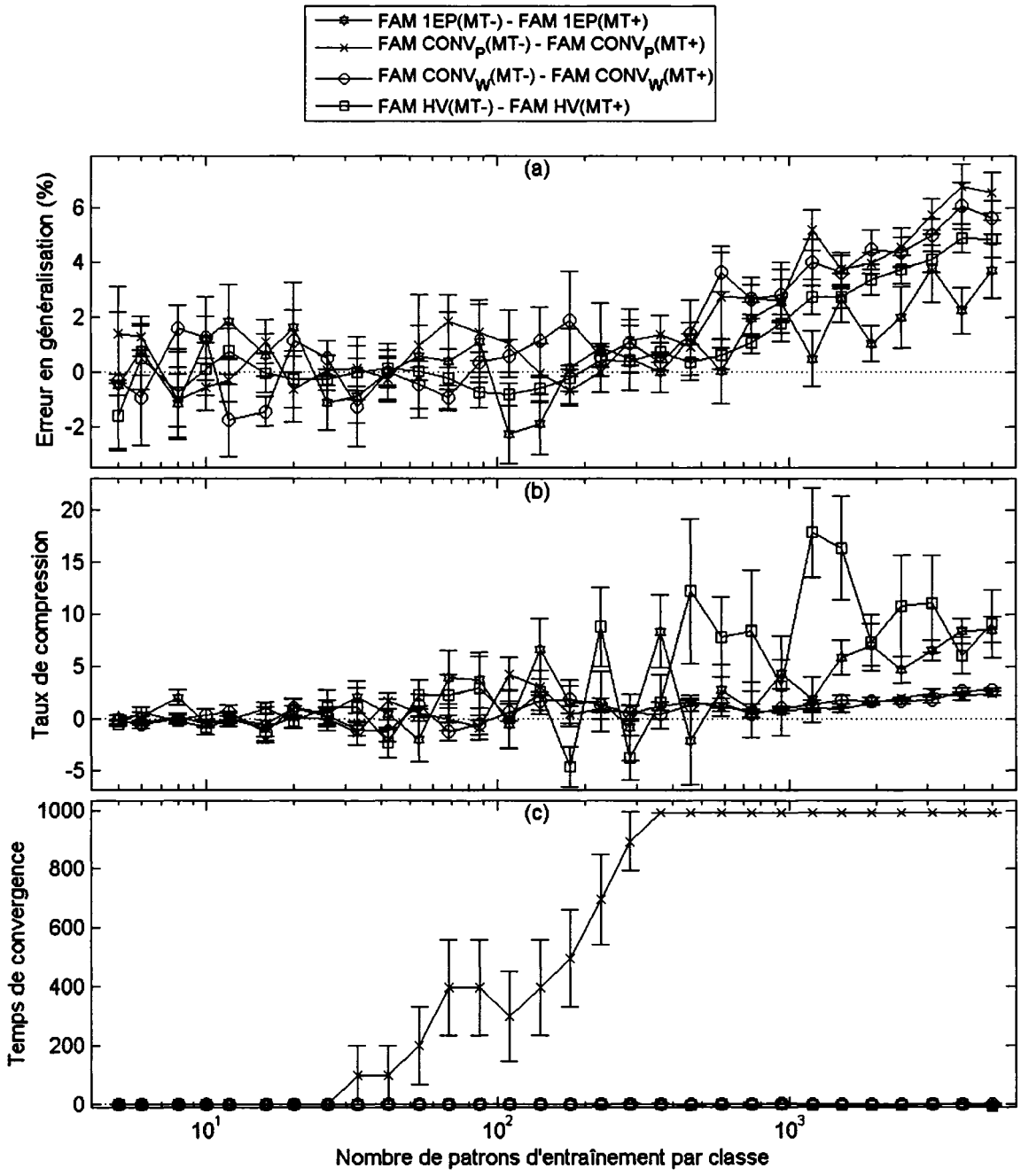


Figure 122 Différence des performances du FAM entre MT- et MT+ avec la base DBμ(9%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

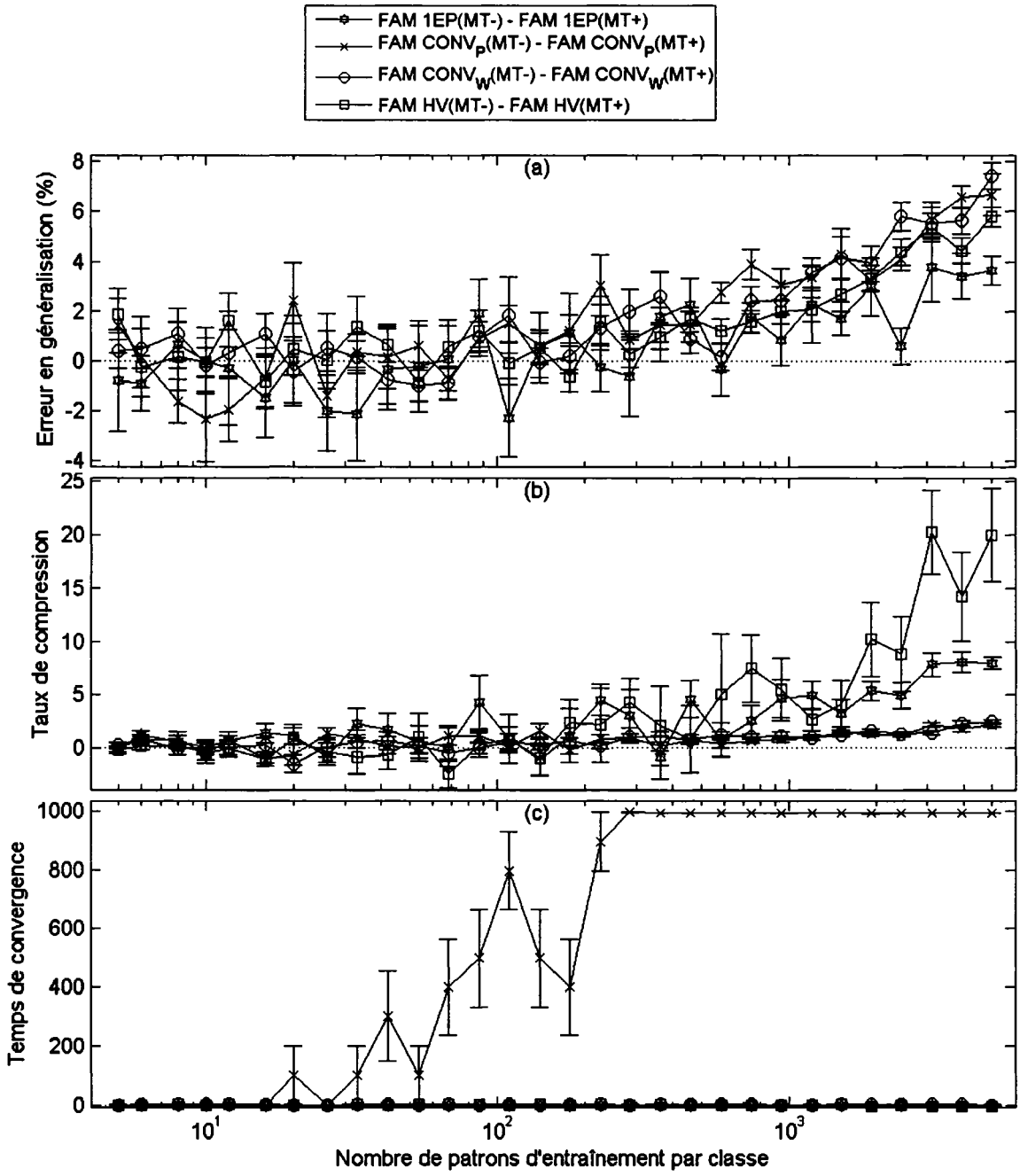


Figure 123 Différence des performances du FAM entre MT- et MT+ avec la base DBμ(11%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

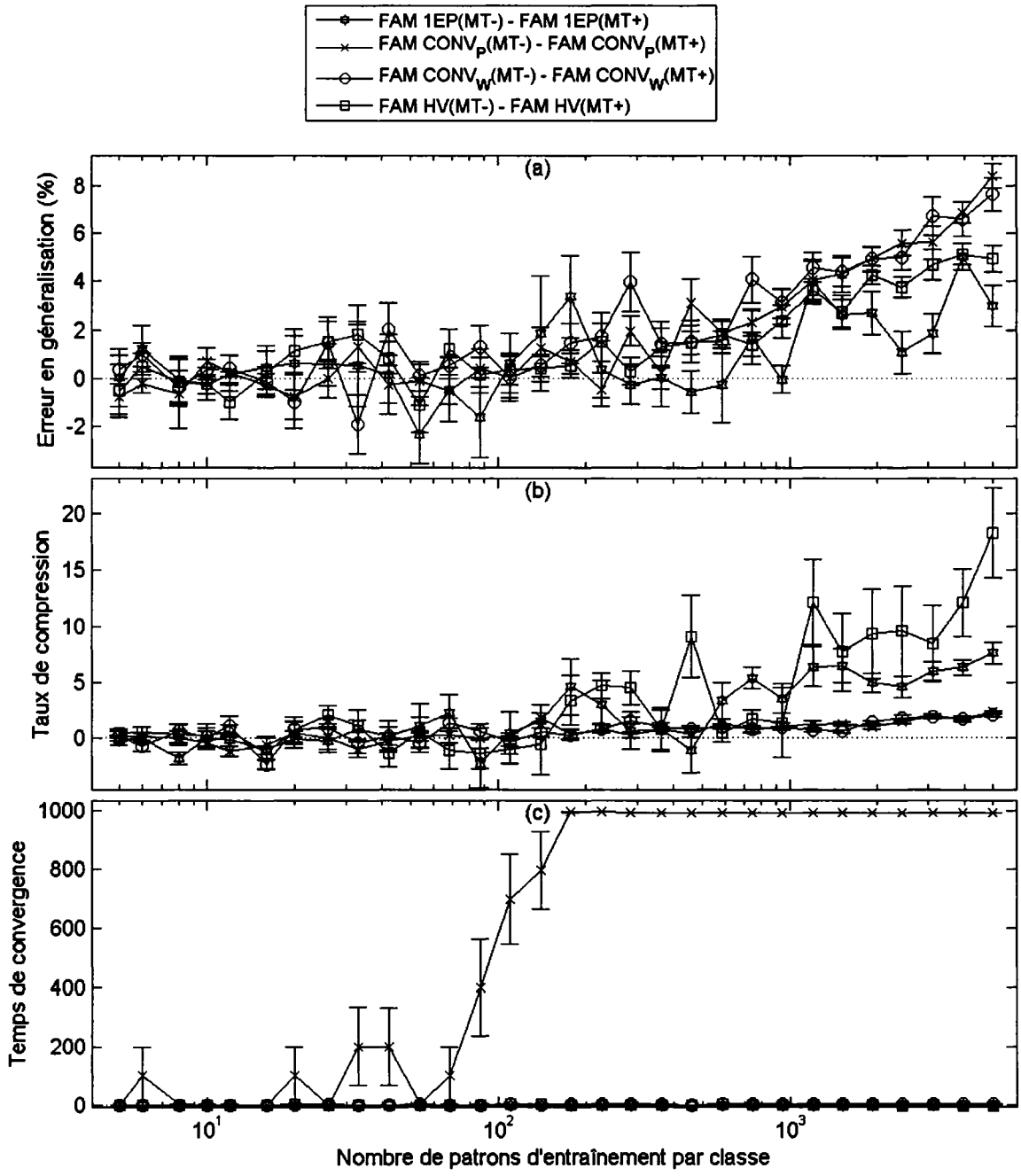


Figure 124 Différence des performances du FAM entre MT- et MT+ avec la base DBμ(13%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

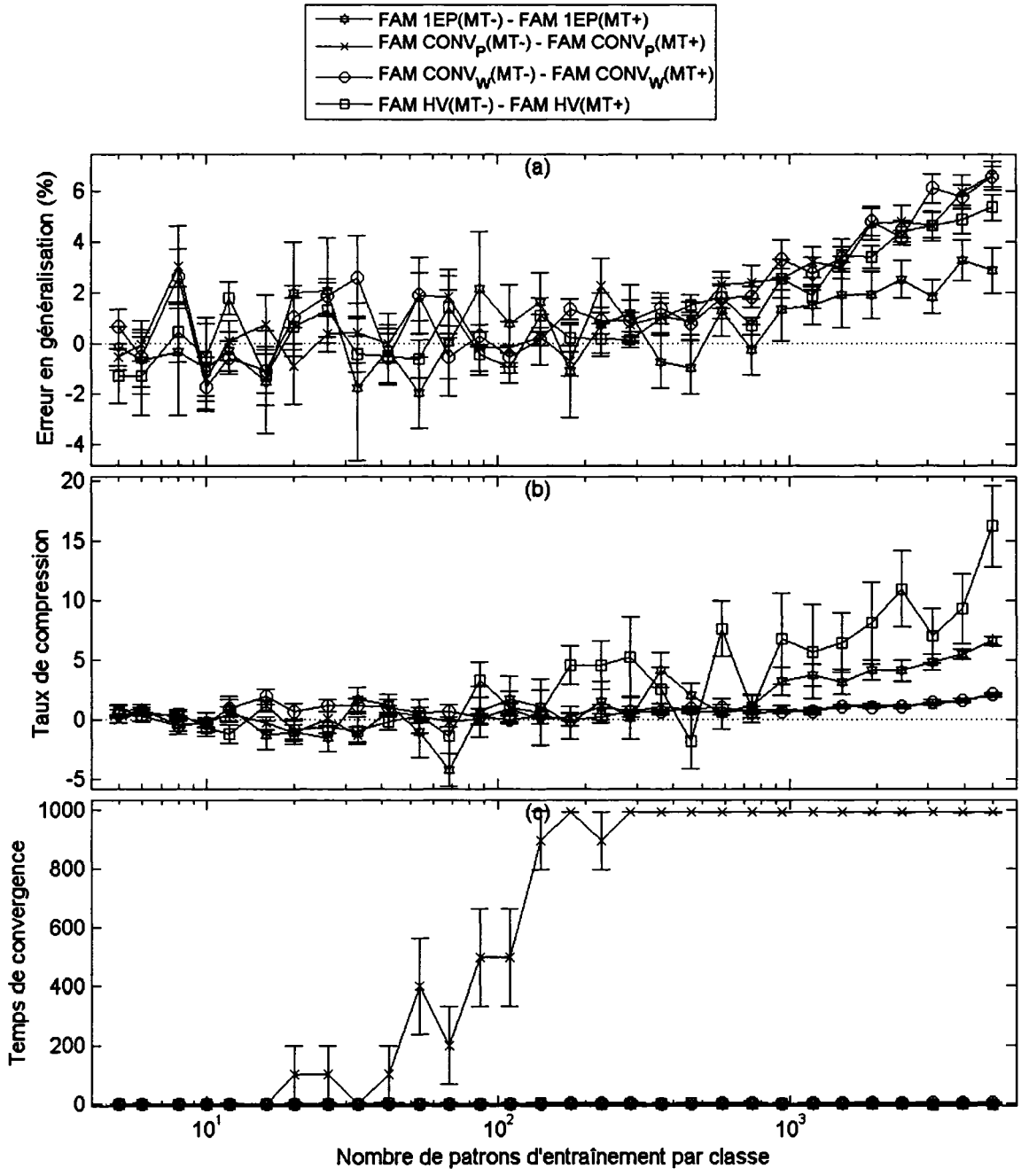


Figure 125 Différence des performances du FAM entre MT- et MT+ avec la base DBμ(15%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

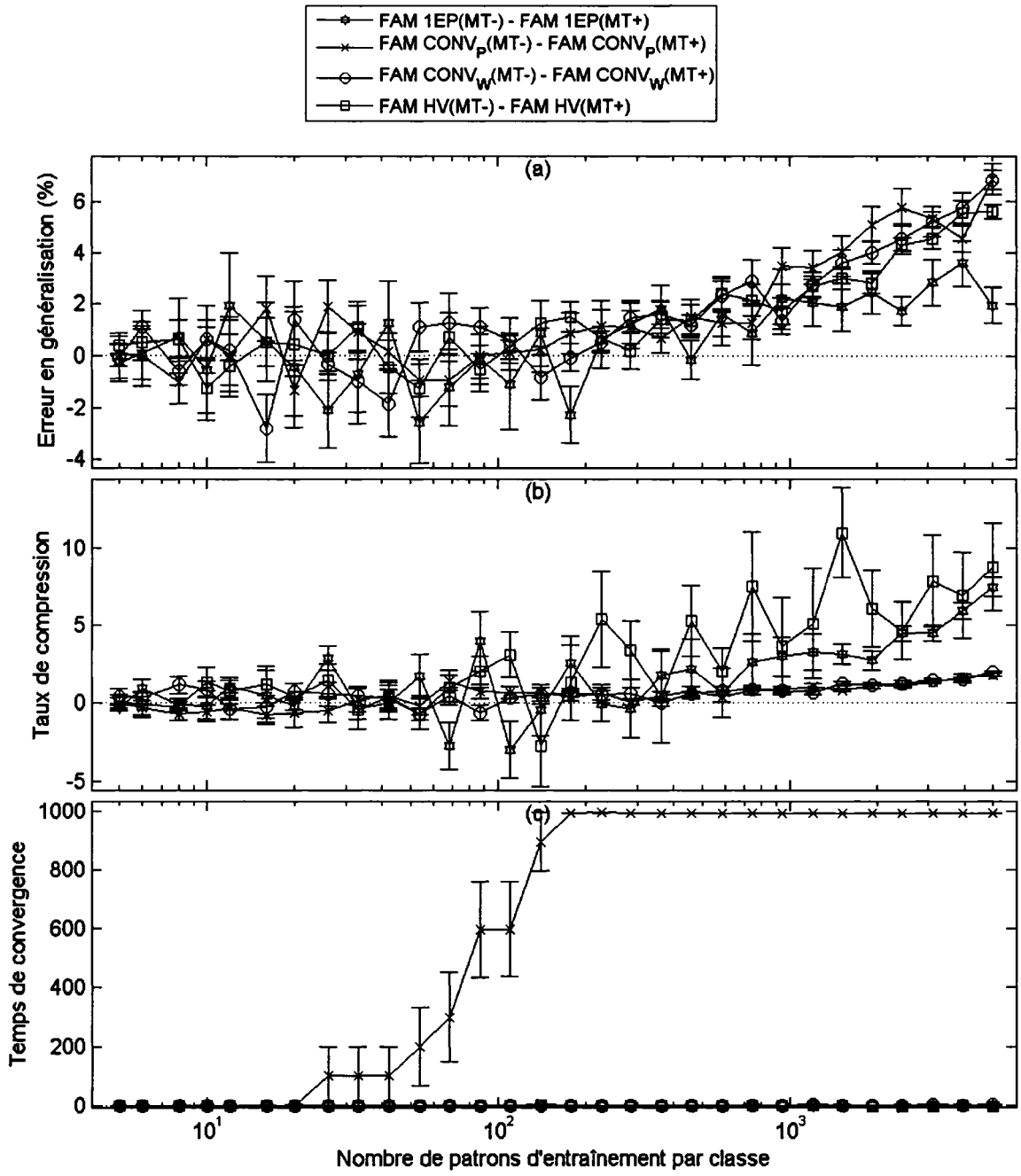


Figure 126 Différence des performances du FAM entre MT- et MT+ avec la base DBμ(17%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

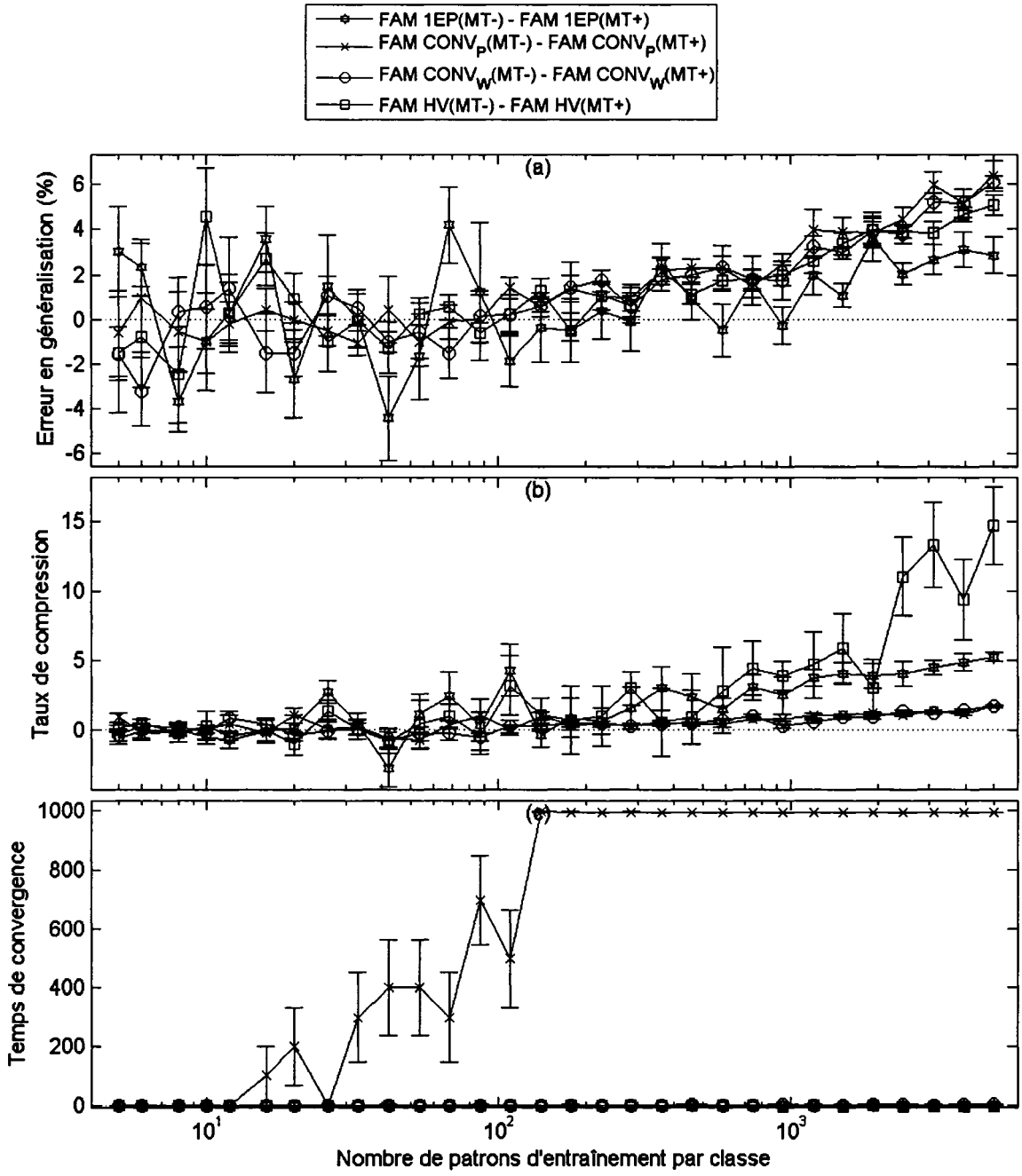


Figure 127 Différence des performances du FAM entre MT- et MT+ avec la base DBμ(19%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

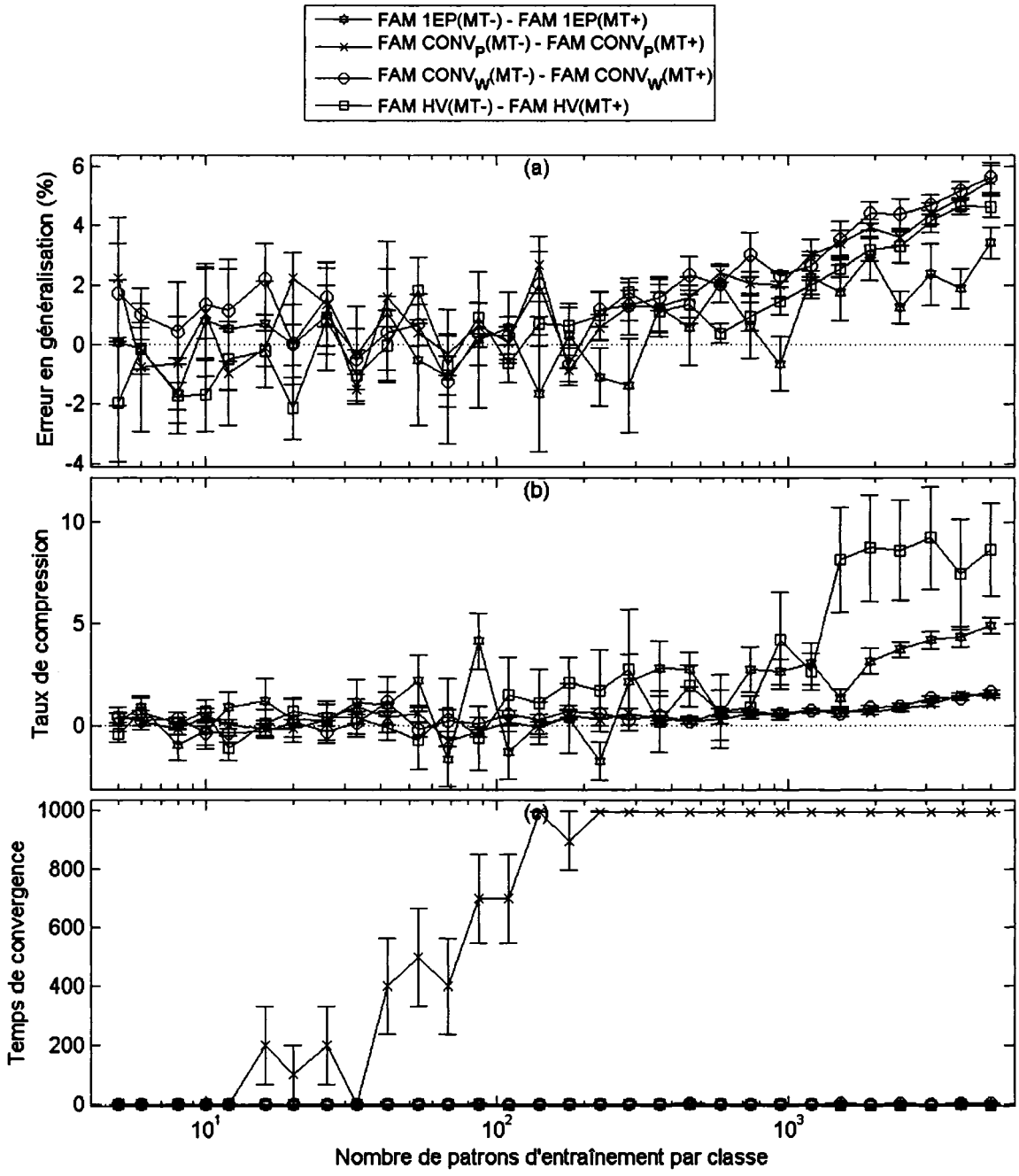


Figure 128 Différence des performances du FAM entre MT- et MT+ avec la base DBμ(21%)
(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

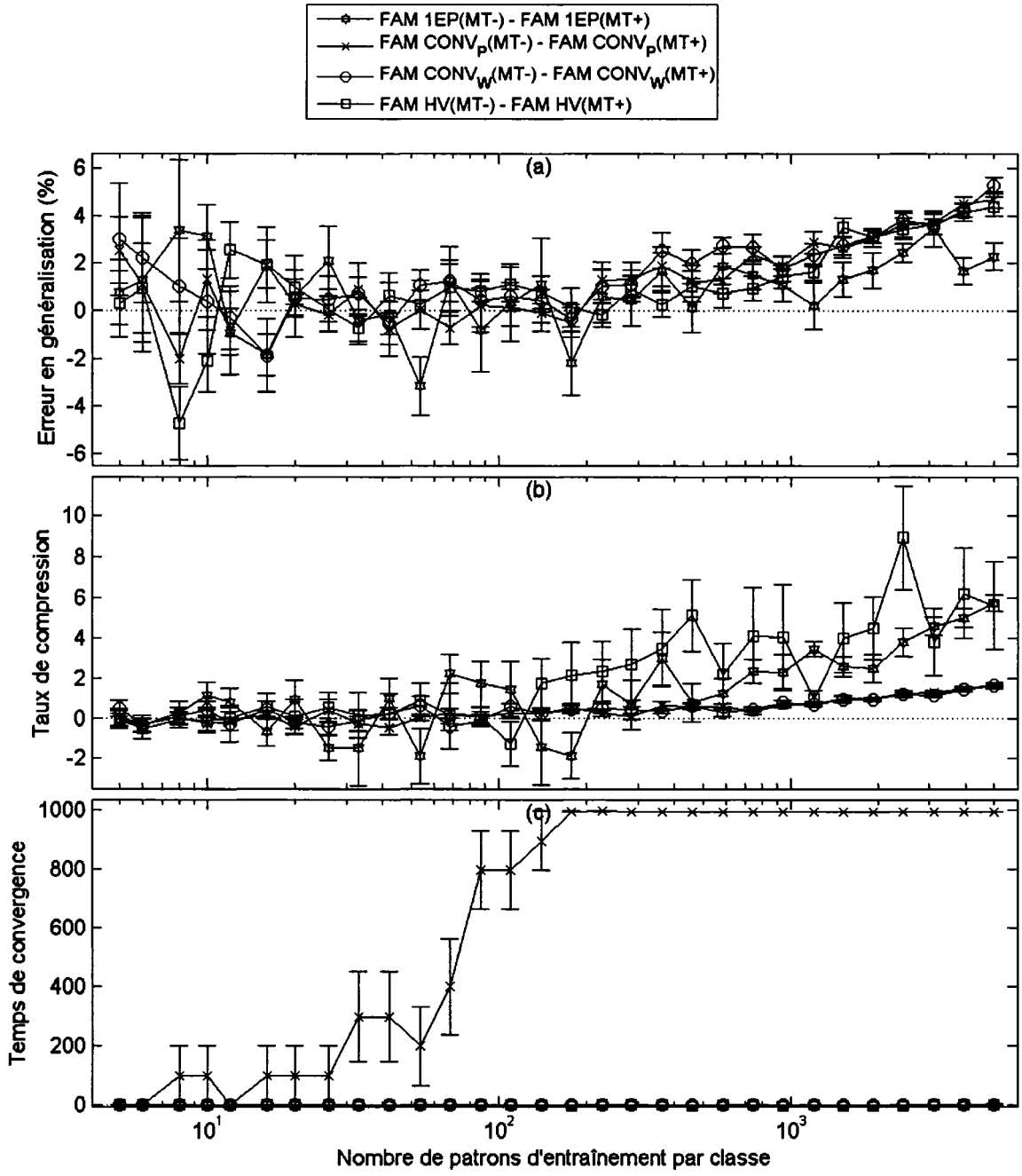


Figure 129 Différence des performances du FAM entre MT- et MT+ avec la base DBμ(23%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

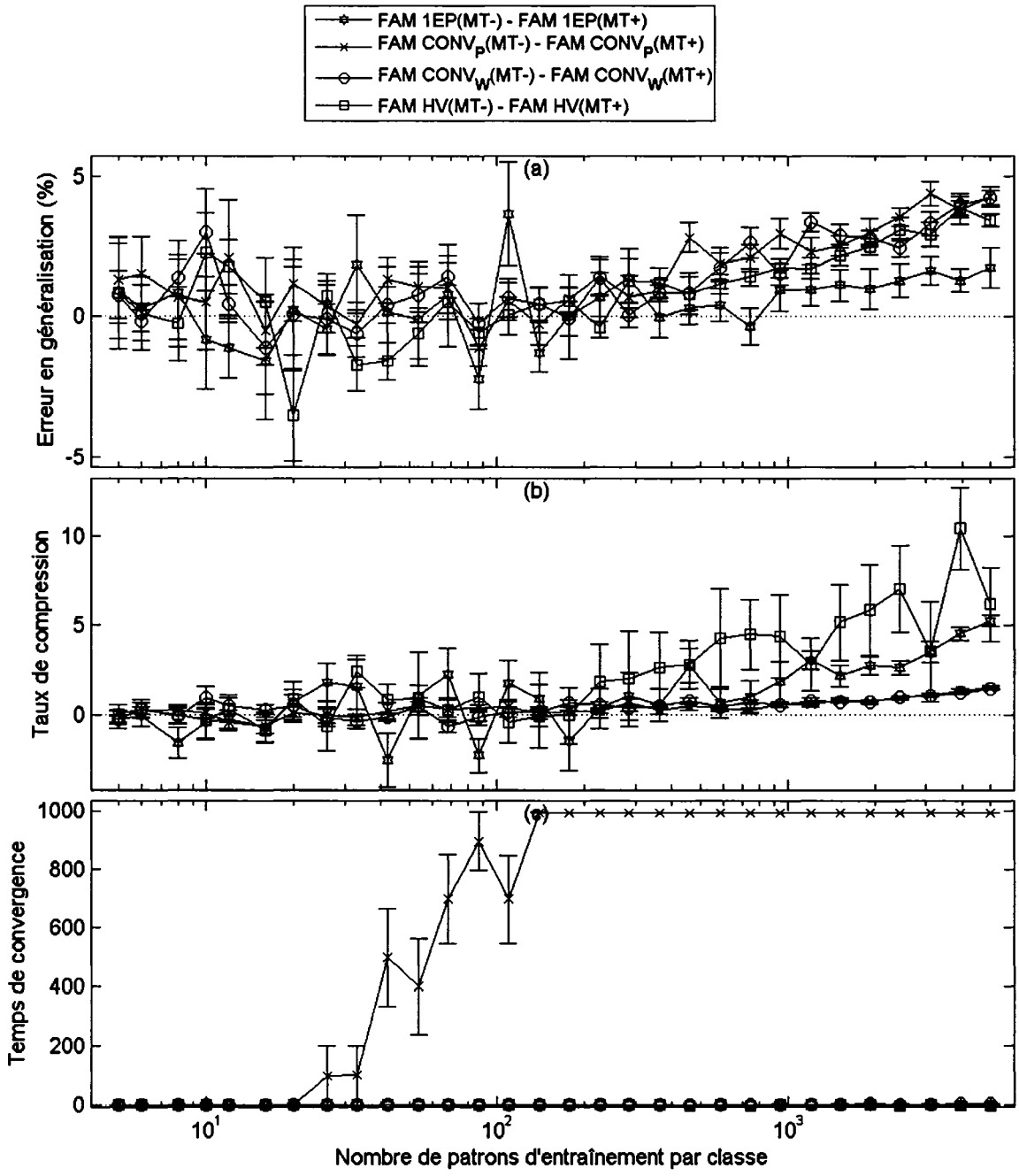


Figure 130 Différence des performances du FAM entre MT- et MT+ avec la base DBμ(25%)

(a) Erreur en généralisation, (b) le taux de compression et (c) le temps de convergence

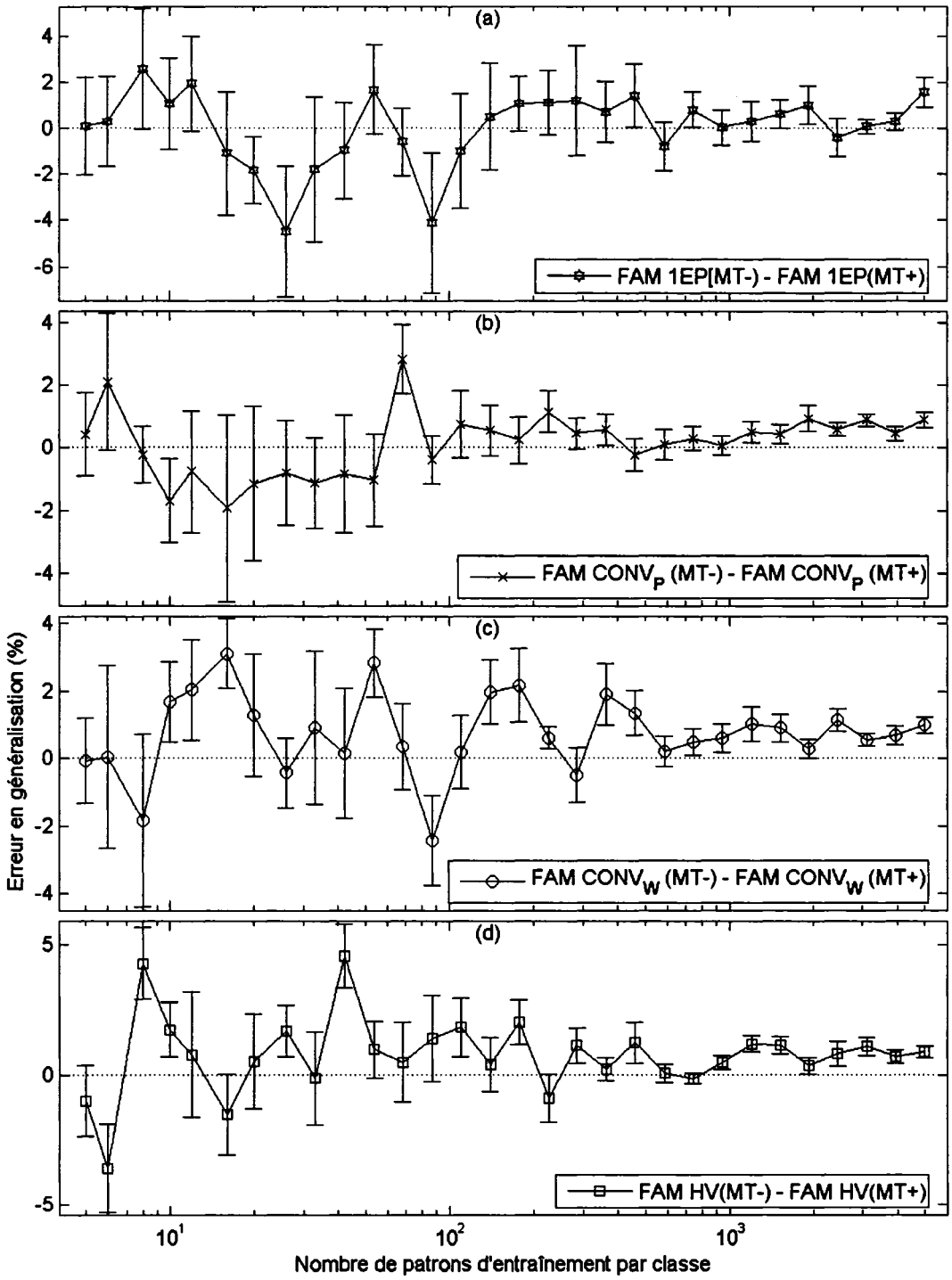


Figure 131 Différence entre l'erreur en généralisation avec la base DB_{P2}
(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

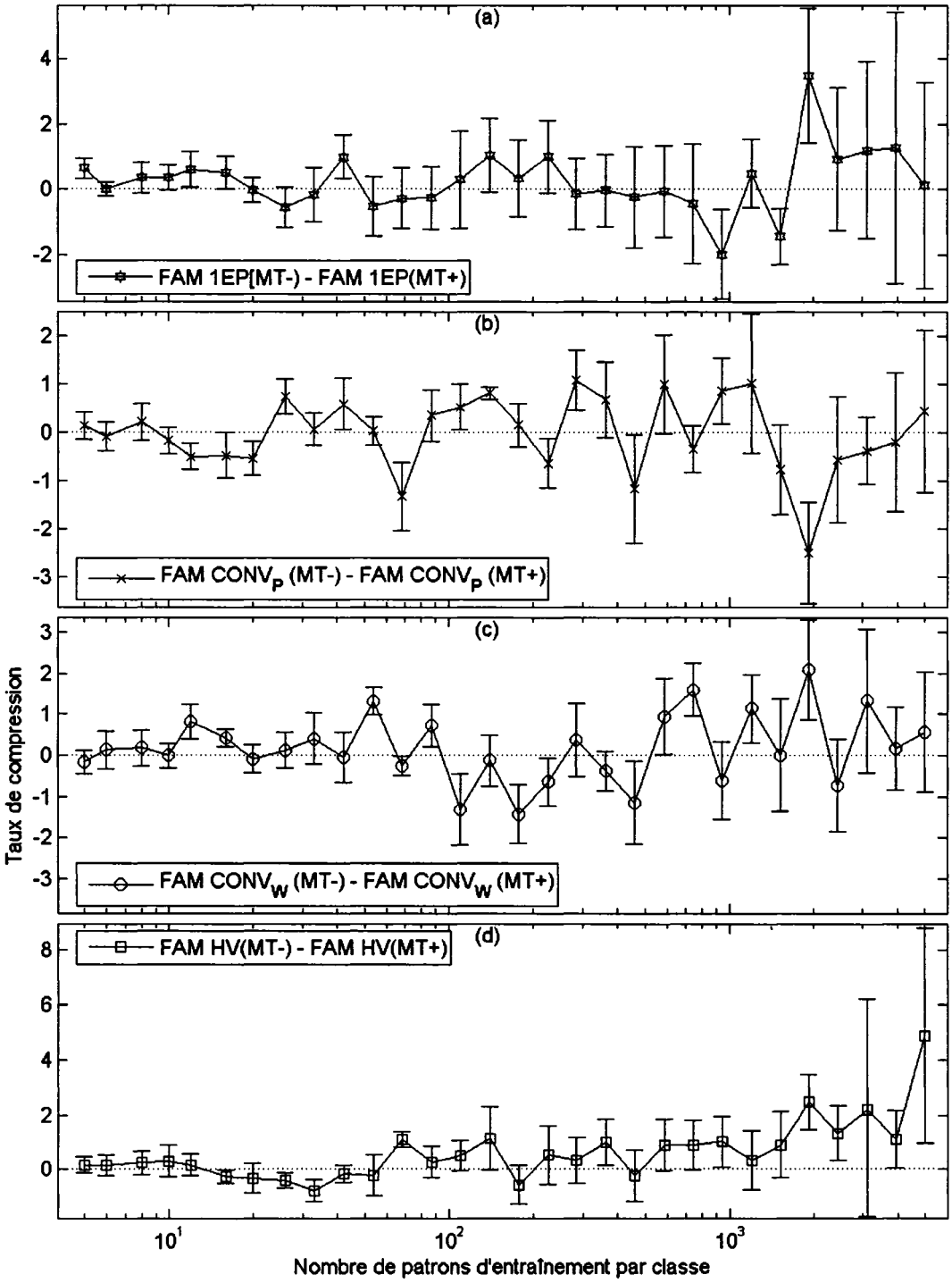


Figure 132 Différence sur le taux de compression avec la base DB_{P2}

(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

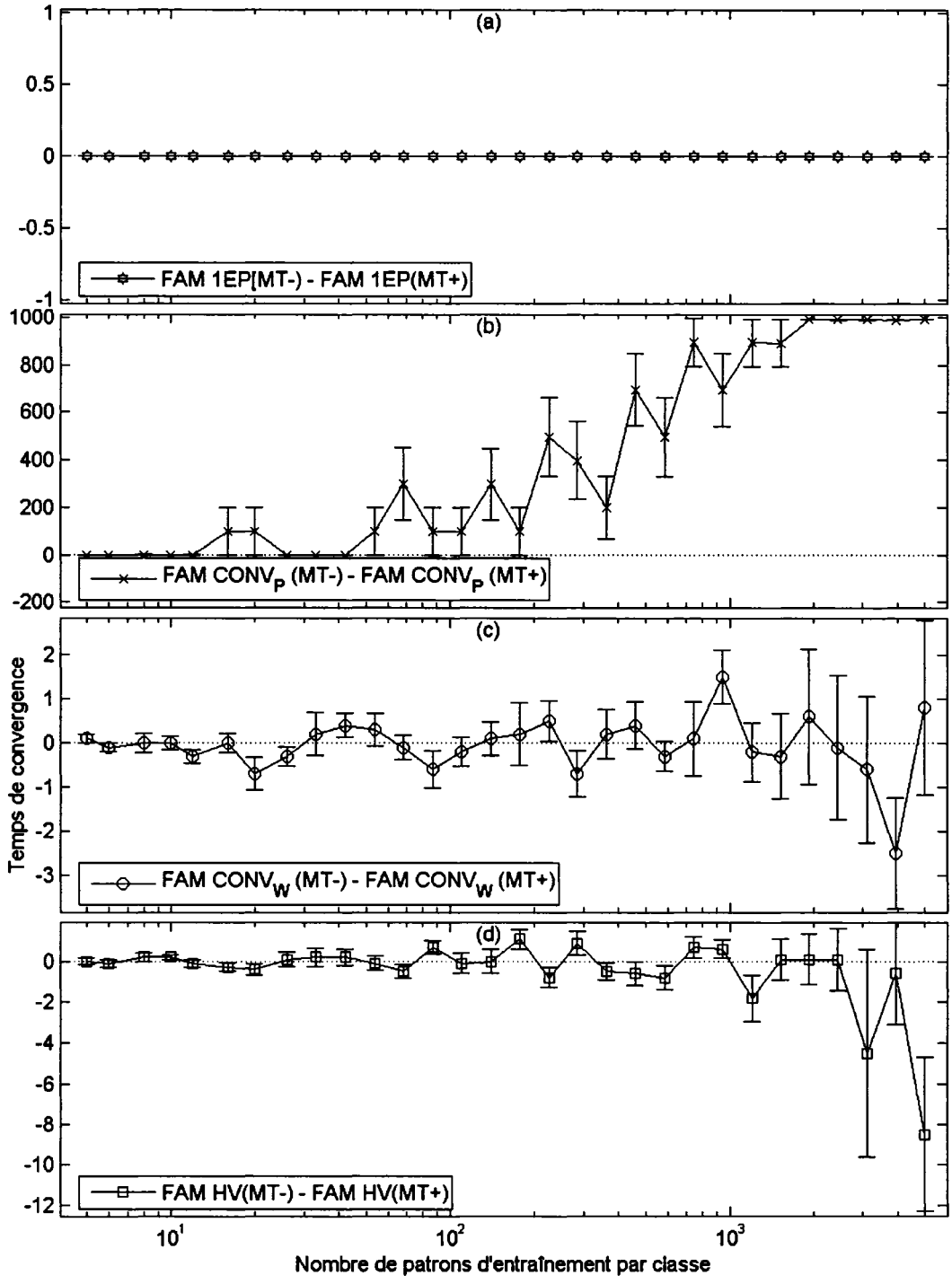


Figure 133 Différence sur le temps de convergence avec la base DB_{P2}

(a) Une époque, (b) Convergence des patrons, (c) Convergence des poids synaptiques et (d) Validation hold-out.

ANNEXE 7

Sommaire des résultats avec les bases de données synthétiques

Tableau XIII

Résultats avec 5k patrons par classe pour la base DB_μ

Stratégies d'apprentissage	Erreur en généralisation moyenne (%) avec 5k patrons par classe				
	DB_μ(1%)	DB_μ (3%)	DB_μ (5%)	DB_μ (7%)	DB_μ (9%)
Erreur théorique	1,00	3,00	5,00	7,00	9,00
CQB	1,00 (0,04)	3,08 (0,05)	4,87 (0,07)	7,00 (0,10)	9,12 (0,08)
***k*NN**	1,08 (0,03)	3,31 (0,06)	5,26 (0,08)	7,48 (0,11)	9,88 (0,08)
1NN	1,54 (0,03)	4,60 (0,11)	7,31 (0,09)	10,36 (0,12)	13,35 (0,10)
FAM 1EP MT-	2,75 (0,20)	8,29 (0,43)	13,81 (0,94)	18,73 (1,04)	22,49 (0,87)
FAM HV MT-	2,17 (0,08)	7,27 (0,36)	11,11 (0,23)	17,02 (0,49)	20,80 (0,56)
FAM $CONV_W$ MT-	2,74 (0,18)	8,43 (0,67)	13,49 (0,85)	17,85 (0,57)	20,45 (0,46)
FAM $CONV_P$ MT-	2,56 (0,09)	8,68 (0,66)	12,20 (0,78)	16,98 (0,43)	22,00 (0,66)
FAM 1EP MT+	2,51 (0,14)	7,49 (0,31)	10,89 (0,37)	14,85 (0,43)	18,78 (0,38)
FAM HV MT+	1,88 (0,05)	5,32 (0,08)	8,59 (0,14)	11,94 (0,18)	15,17 (0,13)
FAM $CONV_W$ MT+	1,97 (0,09)	5,49 (0,09)	8,64 (0,13)	12,05 (0,12)	15,30 (0,16)
FAM $CONV_P$ MT+	1,90 (0,07)	5,49 (0,11)	8,59 (0,13)	12,06 (0,16)	15,44 (0,15)
FAM PSO(1EP)	1,04 (0,03)	3,14 (0,05)	4,99 (0,08)	7,25 (0,13)	9,35 (0,08)

Stratégies d'apprentissage	Erreur en généralisation moyenne (%) avec 5k patrons par classe				
	DB_μ (11%)	DB_μ (13%)	DB_μ (15%)	DB_μ (17%)	DB_μ (19%)
Erreur théorique	11,00	13,00	15,00	17,00	19,00
CQB	11,00 (0,08)	13,16 (0,15)	15,11 (0,15)	16,96 (0,10)	19,25 (0,16)
***k*NN**	11,81 (0,13)	14,27 (0,18)	16,13 (0,13)	18,39 (0,09)	20,71 (0,16)
1NN	15,99 (0,12)	19,07 (0,16)	21,17 (0,15)	23,80 (0,16)	26,72 (0,15)
FAM 1EP MT-	25,17 (0,60)	28,31 (0,69)	30,23 (0,56)	32,07 (0,63)	35,05 (0,64)
FAM HV MT-	23,89 (0,33)	27,31 (0,49)	29,16 (0,33)	32,50 (0,38)	34,33 (0,35)
FAM $CONV_W$ MT-	24,65 (0,51)	28,18 (0,50)	29,64 (0,37)	32,25 (0,35)	35,69 (0,47)
FAM $CONV_P$ MT-	24,79 (0,81)	29,72 (0,55)	30,41 (0,57)	33,28 (0,52)	35,77 (0,61)
FAM 1EP MT+	21,53 (0,50)	25,29 (0,36)	27,34 (0,49)	30,11 (0,18)	32,19 (0,30)
FAM HV MT+	18,08 (0,18)	21,08 (0,14)	23,61 (0,19)	26,28 (0,16)	29,16 (0,18)
FAM $CONV_W$ MT+	18,16 (0,08)	21,23 (0,13)	23,73 (0,19)	26,53 (0,22)	29,26 (0,16)
FAM $CONV_P$ MT+	18,10 (0,10)	21,28 (0,14)	23,80 (0,16)	26,39 (0,23)	29,41 (0,17)
FAM PSO(1EP)	11,16 (0,06)	13,37 (0,14)	15,23 (0,13)	17,33 (0,12)	19,55 (0,16)

Stratégies d'apprentissage	Erreur en généralisation moyenne (%) avec 5k patrons par classe		
	DB_μ (21%)	DB_μ (23%)	DB_μ (25%)
Erreur théorique	21,00	23,00	25,00
CQB	20,97 (0,13)	22,99 (0,12)	25,11 (0,10)
*k*NN	22,70 (0,16)	25,04 (0,13)	27,23 (0,12)
1NN	29,09 (0,12)	31,32 (0,12)	33,49 (0,16)
FAM 1EP MT-	38,19 (0,42)	39,22 (0,47)	40,58 (0,47)
FAM HV MT-	36,99 (0,29)	37,36 (0,34)	39,83 (0,31)
FAM $CONV_W$ MT-	37,16 (0,45)	39,28 (0,39)	40,31 (0,27)
FAM $CONV_P$ MT-	37,12 (0,45)	38,34 (0,32)	40,42 (0,31)
FAM 1EP MT+	34,77 (0,32)	36,93 (0,20)	38,81 (0,36)
FAM HV MT+	31,54 (0,15)	33,81 (0,17)	36,10 (0,20)
FAM $CONV_W$ MT+	31,74 (0,12)	33,69 (0,16)	35,94 (0,15)
FAM $CONV_P$ MT+	31,59 (0,13)	33,68 (0,21)	36,14 (0,20)
FAM PSO(1EP)	21,21 (0,17)	23,21 (0,14)	25,50 (0,09)

Tableau XIV

Résultats avec 5k patrons par classe pour la base DB_σ

Stratégies d'apprentissage	Erreur en généralisation moyenne (%) avec 5k patrons par classe				
	DB_σ(1%)	DB_σ (3%)	DB_σ (5%)	DB_σ (7%)	DB_σ (9%)
Erreur théorique	1,00	3,00	5,00	7,00	9,00
CQB	1,02 (0,04)	2,99 (0,06)	5,00 (0,10)	6,98 (0,11)	9,04 (0,07)
*k*NN	1,10 (0,04)	3,23 (0,07)	5,30 (0,08)	7,40 (0,17)	9,66 (0,07)
1NN	1,61 (0,06)	4,49 (0,07)	7,38 (0,07)	10,33 (0,11)	13,04 (0,14)
FAM 1EP MT-	3,27 (0,56)	9,15 (0,72)	12,52 (0,81)	17,30 (0,54)	22,62 (0,84)
FAM HV MT-	2,15 (0,08)	7,10 (0,37)	12,69 (0,39)	16,67 (0,50)	20,13 (0,43)
FAM $CONV_W$ MT-	2,42 (0,20)	8,72 (0,50)	13,25 (0,46)	16,78 (0,27)	22,07 (0,68)
FAM $CONV_P$ MT-	3,11 (0,43)	8,93 (0,58)	13,68 (0,59)	17,02 (0,64)	20,79 (0,86)
FAM 1EP MT+	2,51 (0,12)	7,44 (0,15)	11,00 (0,33)	14,93 (0,37)	17,98 (0,29)
FAM HV MT+	1,84 (0,05)	5,36 (0,11)	8,75 (0,14)	11,93 (0,21)	14,99 (0,12)
FAM $CONV_W$ MT+	1,97 (0,10)	5,43 (0,09)	8,96 (0,10)	12,11 (0,14)	14,79 (0,10)
FAM $CONV_P$ MT+	1,86 (0,08)	5,51 (0,14)	8,67 (0,08)	11,89 (0,15)	14,90 (0,06)
FAM PSO(1EP)	1,06 (0,04)	3,03 (0,06)	4,99 (0,08)	7,03 (0,12)	9,18 (0,06)

Stratégies d'apprentissage	Erreur en généralisation moyenne (%) avec 5k patrons par classe				
	DB_σ (11%)	DB_σ (13%)	DB_σ (15%)	DB_σ (17%)	DB_σ (19%)
Erreur théorique	11,00	13,00	15,00	17,00	19,00
CQB	11,14 (0,09)	13,07 (0,09)	15,13 (0,10)	17,03 (0,13)	19,05 (0,10)
*k*NN	11,98 (0,10)	14,06 (0,10)	16,50 (0,12)	18,44 (0,09)	20,73 (0,16)
1NN	15,91 (0,12)	18,65 (0,14)	21,67 (0,11)	23,92 (0,14)	26,56 (0,13)
FAM 1EP MT-	25,38 (0,88)	27,64 (0,80)	32,24 (0,65)	33,53 (0,40)	35,56 (0,52)
FAM HV MT-	23,69 (0,43)	26,87 (0,28)	29,09 (0,51)	32,35 (0,34)	34,42 (0,35)
FAM $CONV_W$ MT-	25,25 (0,66)	28,56 (0,68)	31,18 (0,51)	32,34 (0,32)	34,44 (0,42)
FAM $CONV_P$ MT-	24,79 (0,39)	28,38 (0,52)	31,05 (0,48)	33,78 (0,19)	33,86 (0,39)
FAM 1EP MT+	21,77 (0,25)	24,30 (0,25)	28,57 (0,38)	30,28 (0,25)	32,65 (0,31)
FAM HV MT+	18,07 (0,17)	20,87 (0,15)	23,77 (0,18)	26,55 (0,14)	28,99 (0,15)
FAM $CONV_W$ MT+	18,16 (0,16)	20,84 (0,15)	23,97 (0,14)	26,53 (0,12)	28,97 (0,21)
FAM $CONV_P$ MT+	18,03 (0,12)	21,13 (0,16)	23,85 (0,18)	26,30 (0,18)	29,11 (0,09)
FAM PSO(1EP)	11,34 (0,11)	13,24 (0,09)	15,39 (0,08)	17,41 (0,12)	19,39 (0,11)

Stratégies d'apprentissage	Erreur en généralisation moyenne (%) avec 5k patrons par classe		
	DB_σ (21%)	DB_σ (23%)	DB_σ (25%)
Erreur théorique	21,00	23,00	25,00
CQB	21,17 (0,13)	23,14 (0,15)	25,11 (0,12)
*k*NN	22,94 (0,12)	25,08 (0,24)	27,36 (0,11)
1NN	28,96 (0,08)	31,17 (0,18)	33,86 (0,13)
FAM 1EP MT-	37,67 (0,49)	39,83 (0,28)	41,39 (0,48)
FAM HV MT-	36,84 (0,34)	38,20 (0,34)	40,06 (0,16)
FAM $CONV_W$ MT-	37,43 (0,54)	38,18 (0,27)	40,19 (0,43)
FAM $CONV_P$ MT-	36,95 (0,47)	38,61 (0,44)	39,85 (0,16)
FAM 1EP MT+	35,04 (0,30)	36,55 (0,29)	38,86 (0,23)
FAM HV MT+	31,46 (0,19)	33,64 (0,24)	36,52 (0,24)
FAM $CONV_W$ MT+	31,52 (0,17)	33,75 (0,21)	36,08 (0,20)
FAM $CONV_P$ MT+	31,60 (0,19)	33,68 (0,17)	36,33 (0,23)
FAM PSO(1EP)	21,42 (0,13)	23,44 (0,20)	25,55 (0,11)

Tableau XV

Résultats avec 5k patrons par classe pour la base DB_{CIS} et DB_{P2}

Stratégies d'apprentissage	Erreur en généralisation moyenne (%) avec 5k patrons par classe	
	DB_{CIS}	DB_{P2}
Erreur théorique	0,00	0,00
CQB	NA	NA
***k*NN**	0,86 (0,03)	1,65 (0,04)
1NN	0,84 (0,02)	1,61 (0,04)
FAM 1EP MT-	4,20 (0,25)	8,89 (0,44)
FAM HV MT-	1,69 (0,07)	4,26 (0,16)
FAM $CONV_W$ MT-	1,77 (0,09)	4,48 (0,38)
FAM $CONV_P$ MT-	1,59 (0,04)	4,51 (0,19)
FAM 1EP MT+	3,98 (0,21)	7,33 (0,33)
FAM HV MT+	1,58 (0,05)	3,68 (0,07)
FAM $CONV_W$ MT+	1,64 (0,05)	3,66 (0,08)
FAM $CONV_P$ MT+	1,47 (0,05)	3,61 (0,08)
FAM PSO(1EP)	1,35 (0,12)	2,05 (0,10)

BIBLIOGRAPHIE

[1] Carpenter, G.A., Grossberg, S., and Reynolds, J.H. (1991). Supervised Real Time Learning and Classification by a self-organizing. *Neural Networks*, 4, 565-588.

[2] Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H. and Rosenb, D.B. (1992). Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multi-dimensional maps. *IEEE Transactions on Neural Networks*, 3 (5), 698-713.

[3] Carpenter, G.A., Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics and Image Processing*, vol. 37, 54-115.

[4] Carpenter, G.A., Grossberg, S. (1991). *Pattern recognition by Self-Organizing Neural Networks*. Cambridge, MA: MIT press.

[5] Carpenter, G. A., Grossberg, S., & Reynolds, J. H. (1991). Fuzzy ART: An adaptive resonance algorithm for rapid, stable classification of analog patterns. *In International Joint Conference on Neural Networks, IJCNN'91*, 2, 411-416.

[6] Carpenter, G.A., Grossberg, S., and Rosen, D.B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4, 759-771.

[7] Zadeh, L.A. (1965). Fuzzy Sets. *Information and Control*, 8, 338-353.

[8] Pal, S. and Dutta Majumder, D.K. (1981). *Fuzzy Mathematical Approach to pattern recognition*, New York: Plenum Press.

[9] Stone, M. (1977). Asymptotics for and against cross-validation. *Biometrica*, 64(1), 39-35 .

[10] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, B. 2, 111-47.

[11] Reynolds, C.W. (1987). Flocks, herbs and schools : a distributed behavioral model. *Computer Graphics*, 21(4), 25-34.

[12] Heppner, F. and Grenander, U. (1990). A stochastic nonlinear model for coordinated bird flocks. *In S. Krasner, Ed., The Ubiquity of Chaos. AAAS Publications, Washington, DC*, 233-238.

[13] Wilson, E.O. (1975). *Sociobiology: The new synthesis*. Cambridge, MA: Belknap Press.

[14] Kennedy, J., and Eberhart, R. C (1995). Particle swarm optimization. *IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ*, IV, 1942-1948.

[15] Kennedy J. & Eberhart R.C. (2001). *Swarm Intelligence*. Morgan Kaufman Publishers.

[16] Eberhart R.C. & Shi Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. *International Congress on Evolutionary Computation, San Diego, California, IEEE Service Center, Piscataway, NJ*, 84-88.

[17] Geisser, S. (1975). The Predictive Sample Reuse Method with Applications. *Journal of The American Statistical Association*, 70, 320-328.

[18] Lerner, B. and Vigdor, B. (2004). An empirical study of fuzzy ARTMAP applied to cytogenetics, *23rd IEEE Convention of Electrical & Electronics Engineers in Israel*, 301-304.

[19] Carpenter, G. and Tan A.H. (1993). Rules extractions , Fuzzy ARTMAP, and medical databases Proceedinds. W*orld Congress on Neural Networks, Prtland OR, Hillsdale, NJ: Lawrence Erlbaun Associates, vol.I*, 501-506.

[20] Lin, T.H. and Soo, V.W. (1997). Pruning Fuzzy ARTMAP Using the Minimum Description Length Principle in Learning Clinical Databases. I*EEE International Conference on Tools for Artificial Intelligence*, 396.

[21] Blume, M., Van Blerkom, D. A., Esener, S. C. (1996). Fuzzy ARTMAP modifications for Intersecting Class Distributions. *Proceedings of the World Congress on Neural Networks.*

[22] Koufakou, A., Georgiopoulos, M., Anagnostopoulos, G., and Kasparis, T. (2001). Cross-Validation in Fuzzy ARTMAP for large databases. *Neural Networks*, 14, 1279-1291.

[23] Vertzi, S., Heileman, G.L., Georgiopoulos, M., and Healy, M..J. (2001). Boosting the performance of ARTMAP. *Neural Networks*, 14, 1279-1291.

[24] Cover, T. M., and Hart, P. E. (1967). Nearest neigbor pattern classification. *IEEE Transaction on Information Theory*, IT-13(1), 123–140.

[25] B.V. Dasarathy (1973). *Nearest neighbour(NN) norms: NN pattern classification techniques*. IEEE Computer Society Press.

[26] Georgiopoulos, M., Koufakou, A., Anagnostopoulos, G. and Kasparis, T. (2001). Overtraining in Fuzzy ARTMAP: Myth or Reality. *IEEE Transactions on Neural Networks*, vol. 2, 1186-1190.

[27] Shi, Y, and Eberhart, R.C. (1998). A modified particle swarm optimizer. *IEEE International Conference on Evolutionary Computation, Piscataway, NJ: IEEE Press*, 69-73.

[28] Valentini, G. (2003). *Ensemble methods based on bias–variance analysis*. University of Genova, DISI-TH-2003-June 24, PhD. Thesis

[29] Oliveira, L.S. (2003). *Automatic recognition of handwritten numerical strings*. Ecole de technologie superieure, University of Quebec, PhD thesis.

[30] Goutte, C. (1997). Note on free lunches and cross-validation. *Neural Computation*, 9, 1211-1215.

[31] Bob Duin, Delf University of thechnologie. *PRTools, a Matlab Toolbox for Pattern Recognition*.[En ligne].http://www.ph.tn.tudelft.nl/~bob/PRTOOLS.html (Consulté le 21 mai 2004).

[32] Granger, E., Rubin, M., Grossberg, S., and Lavoie, P. (2001). A what-and-where fusion neural network for recognition and tracking of multiple radar emitters. *Neural Networks*, 14, 325-344.

[33] Duda, R. O., Hart, P. E., and Stork, D. G (2001). *Pattern Classification*. John Wiley and Sons Inc.

[34] Henniges, P., Granger, E. and Sabourin, R. (2005). Factor of Overtraining with Fuzzy ARTMAP Neural Networks. *International Joint Conference on Neural Networks, IJCNN'05*, vol 2., 1075- 1080.

[35] Granger, E., Henniges, P., Sabourin, R., and Oliveira, L.S. (2006). Particle Swarm Optimization with fuzzy ARTMAP. *International Joint Conference on Neural Networks, IJCNN'06*.

[36] Bayes, R. T. (1763). Essay toward solving a problem in the doctrine of chances. *Philos. Trans. R. Soc. London*, 53, 370-418.

[37] Laplace, P.S. (1774). Mémoire sur les suites récurro-récurrentes et sur leur usages dans la théorie des hasards. *Savants étranges,* 6, 353-371. Oeuvres 8, 5-24.

[38] Milgram, J., Chériet, M. and Sabourin, R. (2005). Estimating Accurate Multi-class Probabilities with Support Vector Machines. *International Joint Conference on Neural Networks 2005*, 1906-1911.

[39] Canuto, A.M.P., Santos, A.M. (2004). A Comparative Investigation of the RePART Neural Network in Pattern Recognition Tasks. *International Joint Conference on Neural Networks 2004*, 3, 2373-2378.

[40] Canuto, A.M.P., Howells, G., Fairhurst, M. (2000). An Investigation of the Effects of Variable Vigilance within RePART Neuro-Fuzzy Network. *Journal of Intelligent and Robotic Systems - Special Issue on Neural Fuzzy Systems*, 29(4), 317-334.

[41] Dubrawski, A. (1997). Stochastic validation for automated tuning of neural network's hyper-parameters. *Robotics and Autonomous Systems*, 21(1), 83-93.

[42] Dubrawski, A. (1997). Tuning neural networks with stochastic optimization. *Intelligent Robots and Systems*, 2,614-620.

[43] Henniges, P., Granger, E., Sabourin, R. and Oliveira, L.S. (2006). Impact of Fuzzy ARTMAP Match Tracking Strategies on the Recognition of Handwritten Digits. *Artificial Neural Networks In Engineering ANNIE*, soumis.