Evolving Classifier Ensembles Using Dynamic Multi-Objective Swarm Intelligence

Jean-François Connolly, Eric Granger and Robert Sabourin
Laboratoire d'imagerie, de vision et d'intelligence artificielle, École de technologie supérieure
Université du Québec, 1100, rue Notre-Dame Ouest, Montreal, Canada
jfconnolly@livia.etsmtl.ca, {Eric.Granger, Robert.Sabourin}@etsmtl.ca

Keywords: Multi-Classifier Systems, Incremental Learning, Dynamic and Multi-Objective Optimization, ARTMAP Neural Networks, Video Face Recognition, Adaptive Biometrics.

Abstract: Classification systems are often designed using a limited amount of data from complex and changing pattern recognition environments. In applications where new reference samples become available over time, adaptive multi-classifier systems (AMCSs) are desirable for updating class models. In this paper, an incremental learning strategy based on an aggregated dynamical niching particle swarm optimization (ADNPSO) algorithm is proposed to efficiently evolve heterogeneous classifier ensembles in response to new reference data. This strategy is applied to an AMCS where all parameters of a pool of fuzzy ARTMAP (FAM) neural network classifiers, each one corresponding to a PSO particle, are co-optimized such that both error rate and network size are minimized. To sustain a high level of accuracy while minimizing the computational complexity, the AMCS integrates information from multiple diverse classifiers, where learning is guided by the ADNPSO algorithm that optimizes networks according both these objectives. Moreover, FAM networks are evolved to maintain (1) genotype diversity of solutions around local optima in the optimization search space, and (2) phenotype diversity in the objective space. Using local Pareto optimality, networks are then stored in an archive to create a pool of base classifiers among which cost-effective ensembles are selected on the basis of accuracy, and both genotype and phenotype diversity. Performance of the ADNPSO strategy is compared against AMCSs where learning of FAM networks is guided through mono- and multi-objective optimization, and assessed under different incremental learning scenarios, where new data is extracted from real-world video streams for face recognition. Simulation results indicate that the proposed strategy provides a level of accuracy that is comparable to that of using mono-objective optimization, yet requires only a fraction of its resources.

1 INTRODUCTION

In pattern recognition applications, matching is typically performed by comparing query samples against class models designed with reference samples collected a priori from the environment. Biometric template matching is performed with biometric models consisting of a set of one or more templates (reference samples) acquired during an enrollment process, and stored in a gallery. To improve robustness and reduce resources, it may also consists of a statistical representation estimated by training a classifier on reference data. Neural or statistical classifiers may implicitly define a model of some individual's biometric trait by mapping the finite set of reference samples, defined in an input feature space, to an output (scores or decision) space. Since the collection and analysis of reference data is often expensive, these classifiers are often designed using some prior knowledge of the underlying data distributions, user-defined hyperparameters, and a limited number of reference samples.

It is however possible in many biometric applications to acquire new reference samples at some point in time after a classifier has originally been trained and deployed for operations. Labeled and unlabeled samples can be acquired through re-enrollment sessions, post-analysis of operational data, or enrollment of new individuals in the system, allowing for incremental learning of labeled data and semi-supervised learning of reliable unlabeled data (Jain et al., 2006; Roli et al., 2008). Moreover, the physiology of individuals and operational condition may therefore also change over time. In video face recognition, acquisition of faces is subject to considerable variations (*e.g.*, illumination, pose, facial expression, orientation and occlusion) due to limited control over unconstrained operational conditions. In addition, new information, such new individuals, may suddenly emerge, and underlying data distributions may change dynamically (*e.g.*, aging) in the classification environment. Performance may therefore decline over time as facial models deviate from the actual data distribution. Beyond the need for accuracy, efficient classification systems for various real-time applications constitutes a challenging problem. For instance, video surveillance systems use a growing numbers of IP cameras, and must simultaneously process many video feeds, matching facial regions to models.

Some classification systems have been proposed for supervised incremental learning of new labeled data, and provide the means to maintain an accurate and up-to-date model of individuals (Connolly et al., 2008). For example, the ARTMAP and Growing Self-Organizing families of neural network classifiers, have been designed with the inherent ability to perform incremental learning. In addition, some well-known pattern classifiers, such as the Support Vector Machine, the Multi-Layer Perceptron and Radial Basis Function neural networks have been adapted to perform incremental learning. However, the decline of performance caused by knowledge (model) corruption remains a fundamental issue for monolithic classifiers. Techniques in literature are mostly suitable for designing classification systems with an adequate number of samples acquired from ideal and static environments, where class distributions are balanced and remain unchanged over time.

Adaptive ensemble-based techniques like some boosting algorithms (Polikar et al., 2001), may avoid knowledge corruption at the expense of growing system complexity, In these cases, exploiting diversified classifier ensembles has been shown to provide robust and accurate systems (Kuncheva, 2004; Minku et al., 2010; Elwell and Polikar, 2011). A key element in of classifiers ensembles is *classifier diversity measures* (Brown et al., 2005; Minku et al., 2010). Through bias-variance error decomposition, it has been shown empirically that considering diversity for ensemble selection improves the generalization capabilities of multi-classifier systems (Brown et al., 2005).

In previous work, the authors proposed an incremental learning strategy driven by a dynamic particle swarm optimization (DPSO) algorithm to evolve a heterogeneous ensemble of incremental-learning fuzzy ARTMAP (FAM) classifiers. This strategy was applied to an adaptive multi-classifier system (AMCS) for video face recognition, where facial models may be created and updated over time, as new reference data becomes available (Connolly et al., 2012b). In this DPSO-based strategy, each particle corresponds to a FAM network, and the DPSO algorithm co-optimizes all parameters (hyperparameters, weights, and architecture) of a pool of base FAM classifiers such that the error rate is minimized.

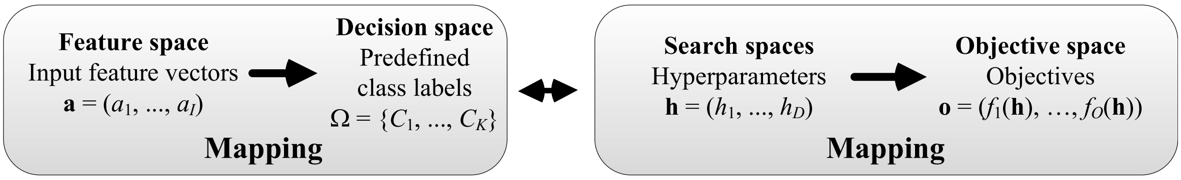
While adaptation was originally performed only according accuracy with mono-objective optimization, the new strategy proposed in this paper is driven by a multi-objective aggregated dynamic niching PSO (ADNPSO) algorithm that also considers the structural complexity of FAM networks during adaptation, allowing to design efficient heterogeneous ensembles of classifiers. The ADNPSO algorithm seeks to maintain both *genotype* diversity (optimization search space) and *phenotype* diversity (optimization search space) among the base classifiers during generation and evolution of pools, and during ensemble selection, according to different criteria. The new ADNPSO incremental learning strategy now optimizes all parameters of a pool of base classifiers such that the error rate and network complexity are minimized. An archive is proposed to store and manage non-dominated classifiers with a Pareto-based criteria. Using local Pareto optimality, networks in the archive are selected to design cost-effective ensembles.

The next section provides motivations for the new incremental learning strategy based on ADNPSO. The strategy (including ADNPSO algorithm and specialized archive) proposed evolve ensembles within AMCSs is presented in Section 3. The data bases, incremental learning scenarios, protocol, and performance measures used for proof-of-concept simulations are then described in Section 4, followed by the experimental results and discussion in Section 5. The ADNPSO learning strategy is validated on a face recognition application in which facial models are to be updated. Performance of AMCSs is assessed in terms of error rate and resource requirements for incremental learning of new data blocks from the real-world video data set captured by the Institute of Information Technology of the Canadian National Research Council (IIT-NRC) (Gorodnichy, 2005).

2 CLASSIFICATION AND OPTIMIZATION

2.1 Fuzzy ARTMAP neural network

ARTMAP refers to a family of self-organizing neural network architectures that is capable of fast, stable, on-line, unsupervised or supervised, incremental learning, classification, and prediction. A key feature of these networks is their unique solution to the stability-plasticity dilemma. The fuzzy ARTMAP



(a) Classification environment

(b) Optimization environment

Figure 1: Pattern classification systems may be defined according to two environments. A *classification environment* that maps a $\mathbb{R}^I$ input feature space to a decision space, respectively defined by feature vectors $\mathbf{a}$, and a set of class labels C_k. As the FAM learning dynamics are govern by hyperparameter vector $\mathbf{h}$, the latter interacts with an *optimization environment*, where each value of $\mathbf{h}$ indicates a position in several search spaces, each one defined by an objective considered during the learning process. For several objective functions (*i.e.*, search spaces), each solution can be projected in an objective space.

(FAM) integrates the unsupervised fuzzy ART neural network to process both analog and binary-valued input patterns into the original ARTMAP architecture (Carpenter et al., 1992). Matching query samples $\mathbf{a}$ against a biometric model of individuals enrolled to a biometric recognition system is typically the bottleneck, especially as the number of individuals grows. The FAM classifier is widely used because it can perform supervised incremental learning of limited data for fast and efficient matching (Granger et al., 2007). Biometric models are learned during training by estimating the FAM weights, architecture and hyperparameters of each individual (*i.e.*, output class) enrolled to the system.

Its architecture consists of three layers: (1) an input layer F_1 of $2I$ neurons, with two neurons associated with each input feature in $\mathbb{R}^I$, (2) a *growing* competitive layer F_2 of J neurons that are each associated to an hyper-rectangle shaped prototype category in the feature space, and (3) a map field F^{ab} of K binary output neurons, each one corresponding to an output class. During supervised learning, FAM grows its F_2 competitive layer to learn an arbitrary mapping between a finite set of training patterns and their corresponding binary supervision patterns. It does so by adjusting its synaptic weights to (1) create and grow the hyper-rectangle shaped prototype categories in the $\mathbb{R}^I$ feature space to perform clustering of the available training samples, and (2) associate those categories (*i.e.*, clusters) to the respective class of the training patterns. When a query pattern is presented to the network, it is propagated through the input F_1 layer and activates a winning F_2 node. The result is a prediction in the form of a binary vector of K outputs where $k = 1$ corresponds to the class label of the winning F_2 node and zero elsewhere.

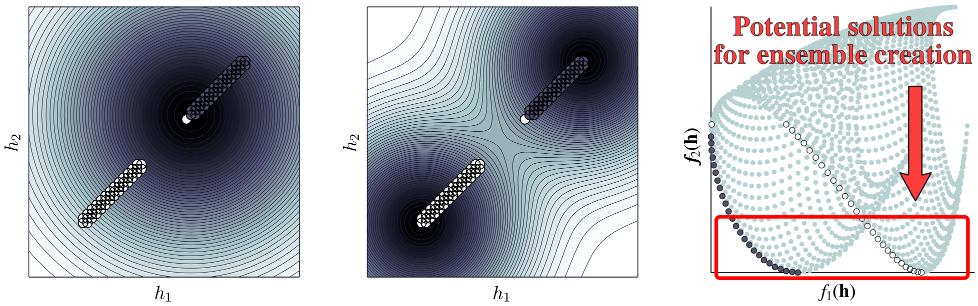
FAM learning dynamics are governed by four hyperparameters: the choice parameter $\alpha \geq 0$, the learning parameter $\beta \in [0,1]$, the match tracking parameter $\varepsilon \in [-1,1]$, and the baseline vigilance parameter $\bar{\rho} \in [0,1]$. Let $\mathbf{h} = (\alpha, \beta, \varepsilon, \bar{\rho})$ be defined as the vector of FAM hyperparameters. These are inter-related, and each one has a distinct impact on the structure of the hyper-rectangle shaped recognition category and implicit decision boundaries formed during training. Computational cost needed to operate FAM are proportional its structural complexity. Given I input features and J nodes on the growing F_2 layer, time complexity to predict a class from a query sample is of $O(IJ)$ (Carpenter et al., 1992).

2.2 Ensembles and dynamic MOO

Although classifier ensembles have been shown to improve accuracy and reliability of pattern recognition systems for a wide range of applications, generating an accurate pool of classifiers and selecting an ensemble among that pool that maximizes prediction accuracy are challenging tasks. To increase diversity among classifiers and improve robustness of such ensembles, swarm intelligence can be used to guide classifiers through *representation space traversal* (Brown et al., 2005). As illustrated in Figure 1, swarm intelligence can be used to explore an hyperparameter search space defined according to a classifier. It guides different classifiers that are trained on the same data, but using different learning dynamics, to create a *heterogeneous* swarm of classifiers (Valentini, 2003). The authors have shown that, since the FAM hyperparameters govern its learning dynamics, genotype diversity among solutions in the search space indeed leads to classifier diversity in the feature and decision spaces (Connolly et al., 2012b).

During incremental learning, adapting the FAM classifier's hyperparameters vector $\mathbf{h} = (\alpha, \beta, \varepsilon, \bar{\rho})$ according to accuracy has been shown to correspond to a dynamic mono-objective optimization problem (Connolly et al., 2012a). In this paper however, multi-objective optimization (MOO) is performed in order to maximize FAM accuracy while minimizing its network computational cost, that is:

$$\text{minimize } \mathbf{f}(\mathbf{h},t) := [f_{\text{NS}}(\mathbf{h},t); f_{\text{ER}}(\mathbf{h},t)], \quad (1)$$



(a) Search space for $f_1(\mathbf{h})$ (b) Search space for $f_2(\mathbf{h})$ (c) Objective space

Figure 2: Position of local Pareto fronts in both search spaces an in the objective space. Obtained with a grid, true Pareto-optimal solutions are illustrated by the dark circles and other locally Pareto-optimal solutions with light circles. While the goal in a multi-objective optimization (MOO) is to find the true Pareto-optimal front (dark circles), another goal of the ADNPSO algorithm is to search both search spaces to find solutions that are suitable for classifiers ensembles. For instance, if at a time t, $f_1(\mathbf{h})$ and $f_2(\mathbf{h})$ respectively correspond to $f_s(\mathbf{h},t)$ and $f_e(\mathbf{h},t)$, these would be solutions in the red rectangle in Figures 2c (with low generalization error and for a wide range of FAM network F_2 sizes).

where $f_{\text{NS}}(\mathbf{h},t)$ is the network size (*i.e.*, number of F_2 layer nodes), and $f_{\text{ER}}(\mathbf{h},t)$ is the generalization error rate of the FAM network for a given hyperparameter vector $\mathbf{h}$ and after learning data set D_t at a discreet time t (Connolly et al., 2012a). Although it was not verified, it is assumed in this paper that training FAM with different values of $\mathbf{h}$ leads to different number of FAM F_2 nodes and that the objective function $f_{\text{NS}}(\mathbf{h},t)$ also corresponds to a dynamic optimization problem.

As a MOO problem, the first goal of the optimization module is to find the Pareto-optimal front of non-dominated solutions. Given the set of objectives $\mathbf{o}$ to minimize, a vector $\mathbf{h}^d$ in the hyperparameter space is said to *dominate* another vector $\mathbf{h}$ if:

$$\begin{aligned} &\forall o \in \mathbf{o} : f_o(\mathbf{h}^d) \leq f_o(\mathbf{h}), \text{ and} \\ &\exists o \in \mathbf{o} : f_o(\mathbf{h}^d) < f_o(\mathbf{h}). \end{aligned} \tag{2}$$

The Pareto-optimal set, defining a Pareto front, is the set of non-dominated solutions.

When adapting classifiers through incremental learning, another goal of the optimization algorithm is to seek hyperparameter values that will give a diversified pool of FAM networks among which accurate ensembles may be selected. As illustrated in Figure 2 with a simple MOO problem, the optimization process should provide accurate solutions with different network structural complexities. This results in ensembles with good generalization capabilities, but with a possibility of having a moderate overall computational complexity.

In this particular case, an optimization algorithm should tackle a dynamic optimization problem by considering several objectives, and yield classifiers that corresponds to vectors $\mathbf{h}$ that are not necessarily Pareto-optimal (see Figure 2). Classic DPSO algorithms as well as MOO algorithms, such as NSGA, MOEA, MOSPO, etc. are not well suited for such problems. The only approach in literature that is aimed at generating and evolving a population of FAM networks that are diverse in term of structural complexity, yet contained non-dominated alternatives, is presented in (Li et al., 2010). In this case, a memetic archive was instead used to prune F_2 layer nodes and categorize FAM networks into sub-populations that evolved independently according to some genetic algorithm. With this method, FAM networks must be pruned to maintain phenotype diversity and can only be applied when they are accessible, which is not the case in the present paper.

3 EVOLUTION OF ENSEMBLES

This paper seeks to address challenges related to the design of robust AMCSs, where class models are designed and updated over time as new reference data becomes available. In this section, a new ADNPSO incremental learning strategy – integrating an aggregated dynamical niching PSO (ADNPSO) algorithm and a specialized archive – is described to evolve heterogeneous classifier ensembles in response to new data. Each particle in the optimization environment corresponds to a FAM network in the classification environment, and the ADNPSO learning strategy evolves a swarm of classifiers such that both FAM error rate and network size are minimized.

3.1 Adaptive multi-classifier systems

Figure 3 depicts the evolution of an adaptive multi-classifier system (AMCS) for incremental learning of new data. It is composed of (1) a long term mem-

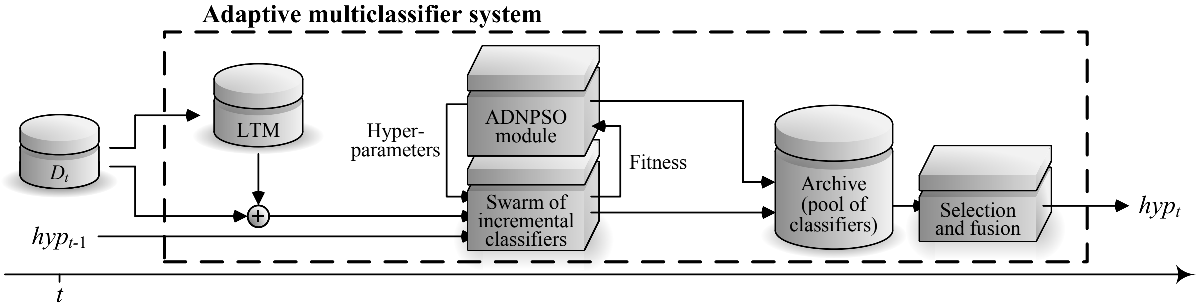


Figure 3: Evolution over time of the adaptive multi-classifier system (AMCS) in a generic incremental learning scenario, where new blocks of data are used to update a swarm of classifiers. Let D_1, D_2, ... be blocks of training data that become available at different instants in time $t = 1, 2, \ldots$. The AMCS starts with an initial hypothesis hyp_0 from prior knowledge of the classification environment. Given a new data blocks D_t, each hypothesis hyp_{t-1} is updated to hyp_t by the AMCS.

ory (LTM) that stores and manages incoming data for validation, (2) a population of base classifiers, each one suitable for supervised incremental learning, (3) a dynamic population-based optimization module that tunes the user-defined hyperparameters of each classifier, (4) a specialized archive to keep a pool of classifiers for ensemble selection, and (5) an ensemble selection and fusion module.

When a new block of learning data D_t becomes available to the system at a discrete time t, it is employed to update the long term memory (LTM), and evolve the swarm of incremental classifiers. The LTM stores data samples from each individual class for validation during incremental learning and fitness estimation of particles on the objective function (Connolly et al., 2012a). Each particle in the search spaces are associated to a FAM network, and the ADNPSO module, through a learning strategy, co-jointly determines the classifiers hyperparameters, architecture, and parameters such that FAM networks error rate and size are minimized. A specialized archive stores a pool of classifiers, corresponding to locally non-dominated solutions (of different structural complexity) found during the optimization process. Once the optimization process is complete, the selection and fusion module produces an heterogeneous ensemble that is combined with a simple majority vote.

3.2 Aggregated dynamical niching PSO

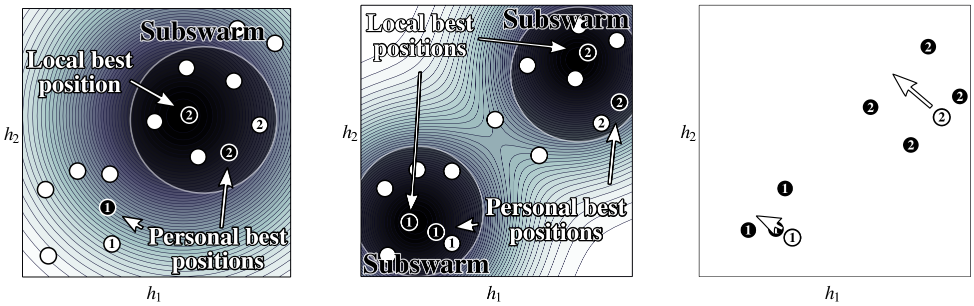
Particle swarm optimization (PSO) is a swarm intelligence stochastic optimization technique that is inspired by social behavior of bird flocking or fish schooling. With PSO, each particle corresponds to a single solution in the optimization environment, and the population of particles is called a swarm. In a mono-objective problem and at a discreet iteration τ, particles move through the hyperparameter search space and change their positions $\mathbf{h}(\tau)$ under the guidance of different sources of influence. Unlike evolutionary algorithms (such as genetic algorithms), each particle always keeps in memory its best position and the best position of its surrounding. PSO algorithms use this information and move the swarm according to a *social influence* (*i.e.*, their neighborhood previous search experience) and a *cognitive influence* (*i.e.*, their own previous search experience).

To tackle ensemble creation, ADNPSO uses the same approach as mono-objective optimization algorithms and defines influences in the search spaces, with the objective functions. As Figure 2 previously showed with a simple problem, the rational behind this approach is that when several local optima are present in different search spaces, non-dominated solutions tend to be located in regions between local optima of the different objective. Each particle will then move according to a cognitive and social influence for error rate (ER) and network size (NS) objectives (see Figure 4). Formally this is defined by:

$$\begin{aligned}\mathbf{h}(\tau+1) &= \mathbf{h}(\tau) + w_0\ (\mathbf{h}(\tau) - \mathbf{h}(\tau-1)) \\ &+ r_1\ w_1\ (\mathbf{h}_{\text{soc., ER}} - \mathbf{h}(\tau)) + r_2\ w_2\ (\mathbf{h}_{\text{cog., ER}} - \mathbf{h}(\tau)) \\ &+ r_3\ w_3\ (\mathbf{h}_{\text{soc., NS}} - \mathbf{h}(\tau)) + r_4\ w_4\ (\mathbf{h}_{\text{cog., NS}} - \mathbf{h}(\tau)),\end{aligned} \tag{3}$$

During optimization, each particle (1) begins at its current location, (2) continues moving in the same direction it was going according to an inertia weight w_0, and (3) is attracted by each source of influence according to a weight w_θ adjusted by random numbers r_θ. By adjusting the weights w_θ, the swarm may be biased according to one objectives or even divided in three sub-populations : (1) one that specializes in accuracy (w_1 and $w_2 > w_3$ and w_4), (2) one that specializes in complexity (w_1 and $w_2 < w_3$ and w_4), and (3) a generalist subpopulation that put both objectives on equal footing ($w_1 = w_2 = w_3 = w_4$).

Social influences of both objectives are managed by creating subswarms by adapting the DNPSO lo-



(a) Search space for $f_1(\mathbf{h})$ (b) Search space for $f_2(\mathbf{h})$ (c) Particle movements

Figure 4: Examples of influences in the search spaces and resulting movements. Given the same objective functions used in Figure 2, two particles among a swarm (white circles), and their social and cognitive influences (black circles), let subswarms set to have a maximal size of 5 particles. While both particles 1 and 2 are going to have cognitive influences in both search spaces, particle 1 is not part of any subswarm for $f_1(\mathbf{h})$. Unlike particle 2, it has no social influence for this objective and ADNPSO sets $w_1 = 0$ when computing its movement with Equation 3.

cal neighborhood topology (Nickabadi et al., 2008) to multiple objectives. While DNPSO dynamically creates subswarms around the *current position* of local best particles that are particles with a personal best position that has the best fitness in their neighborhood, the ADNPSO rather use the memory of these (local best) particles. Social influences are then *personal best position* of local best particles computed independently for both objectives. As shown in Figure 4, by limiting the size of each subswarm, particles can be excluded of these subswarms for none, one, or both objectives. For the objective for which it is excluded, it is said to be free and its social influence is removed by setting the weights $w_1 = 0$ and/or $w_3 = 0$ when computing Equation 3. This way, poor compromises can be avoided and conflicting influences can then be managed simply by limiting the maximal size of each subswarm.

Although the DNPSO local neighborhood topology insures in many ways particle diversity in the search space, it is also adapted to also maintain cognitive (*i.e.*, personal best) diversity among particles within each subswarm. The ADNPSO algorithm defines a distance Δ around local best positions of each objectives. Every time a particle moves with the distance Δ from the detected local optima of one objective, "loses its memory" for that objective by erasing its personal best value. It then moves only according the other objectives b settings the proper weights to 0.

3.3 Archive and ensemble selection

Since a limited amount of reference data is used to design the AMCS, both objectives are discrete function, and the accuracy (error rate) is prone to over fitting. In this context, a specialized archive is used to (1) insure phenotype diversity in the objective space according to FAM network size, and (2) as framework for ensemble selection.

As Figure 5 shows, it categorizes FAM networks associated with each solution found in the search space according to their network size and stores them to create a pool of classifiers among which ensembles can be selected. Although, for a MOO formulation, this imply keeping dominated solutions in the objective space, using a specialized archive ensures storing classifiers with a wide phenotype diversity for FAM network F_2 layer size.

While a genotype local best topology in the hyperparameter space is used to define neighborhoods and zones of influence for the different particles, the same principle is applied in the objective space for ensemble selection. The most accurate FAM of each network size domain are considered as phenotype local best solutions. Classifiers are selected to create an initial ensemble that is completed with a second selection phase that uses a greedy search process (introduced in (Connolly et al., 2012b)) to increase classifier diversity by maximizing their genotype diversity.

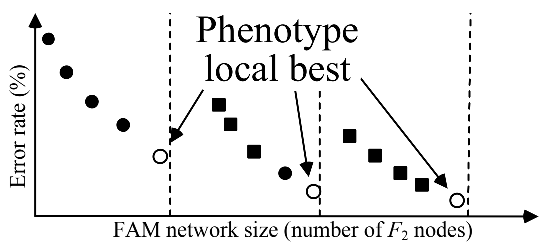


Figure 5: Specialized archive in the objective space. The FAM network size objective is segmented in different domains, where both Pareto-optimal (circles) and locally Pareto-optimal (squares) solutions are stored. The local best are defined as the most accurate network of each domain.

3.4 ADNPSO learning strategy

An ADNPSO incremental learning strategy (Algorithm 1) is proposed to evolve FAM networks according multiple objectives and accumulate a pool of FAM networks in the archive presented in Section 3.3. During incremental learning of a data block D_t, FAM hyperparameters, parameters and architecture are cojointly optimized such that the generalization error rate and network size are minimized. Based on the hypothesis that maintaining diversity among particles in the optimization environment implicitly generates diversity among classifiers in the classification environment (Connolly et al., 2012b), properties of the ADNPSO algorithm is used to evolve a diversified heterogeneous ensembles of FAM networks over time.

At a time t, and for each particle n, the current particle position is noted $\mathbf{h}_n$, along with its personal best values on each objective function o, $\mathbf{h}^*_{n,o}$. The values estimated on the objective functions and the best position of each particle are respectively noted $f_o(\mathbf{h}_n,t)$ and $f_o(\mathbf{h}^*_{n,o},t)$. For O objectives, and the ADNPSO algorithm presented in Section 3.2 that uses N particles, a total of $(O+2)N$ FAM networks are required.

For each particle n, the AMCS stores:

1. O networks $FAM_{n,o}$ associated with $\mathbf{h}^*_{n,o}$ (particle n personal best position on each objective function o),
2. the network FAM_n^{start} associated to the current position of the each particle n after convergence of the optimization process at time $t-1$, and
3. the network FAM_n^{est} obtained after learning D_t with current position of particle n (noted $\mathbf{h}_n$).

While FAM_n^{start} represents the state of the particle before learning D_t, FAM_n^{est} is the state of the same particle after having explored a position in the search space, and it is used for fitness estimation.

During the initialization process (line 1), the swarm and all FAM networks are initialized. Particle positions are randomly initialized within their allowed range. When a new D_t becomes available, the optimization process begins. Networks associated with the best position of each particle ($FAM_{n,o}$) are incrementally updated using D_t, along with their fitness $f_o(\mathbf{h}^*_{n,o},t)$ (lines 3–5). Network in the archive and their fitness are also updated in the same manner (lines 6–13). Since accuracy corresponds to dynamic optimization problem, Algorithm 1 verifies if solutions still respect the non-dominant criteria of the specialized archive. Then, the specialized archive is filled accordingly using networks $FAM_{n,o}$.

Optimization continues were it previously ended until the ADNPSO algorithm converges (lines 14–27). During this process, the ADNPSO algorithm explores the search spaces (line 15). A copy of FAM_n^{start} redefines the state of FAM_{est} prior learning at a time t, trains the latter using $\mathbf{h}_n$, and estimates its fitness (lines 17–19). For each objective o, the best position ($\mathbf{h}^*_{n,o}$) and its corresponding fitness ($f_o(\mathbf{h}^*_{n,o},t)$) and network ($FAM_{n,o}$) are updated if necessary (lines 20–26). If $f_o(\mathbf{h}_n,t)$ and $f_o(\mathbf{h}^*_{n,o},t)$ are equal, the network that requires the least resources (F_2 nodes) is used. Each time fitness is estimated at a particle current position, FAM_{est} is categorize according its network size and added to the archive if it is non-dominated in its F_2 size domain (lines 23–26). Finally, the iteration counter is incremented (line 27). Once optimization converges, networks corresponding to the last position evaluated of every particle (FAM_n^{est}) are stored in FAM_n^{start} (lines 28–29). These networks will define the swarm's state prior learning data block D_{t+1}.

4 EXPERIMENTAL METHODOLOGY

Proof-of-concept simulations focus on classification of facial regions captured in a video face recognition applications. Since matching is perform with an AMCS based on adaptive FAM ensembles, systems responses for each successive query sample is a binary code (equals to "1" for the predicted class, and "0"'s for the others).

The data base was collected by the Institute for Information Technology of the Canadian National Research Council (IIT-NRC) (Gorodnichy, 2005). It is composed of 22 video sequences captured from eleven individuals positioned in front of a computer. For each individual, two color video sequences of about fifteen seconds are captured at a rate of 20 frames per seconds with an Intel web cam of a 160×120 resolution that was mounted on a computer monitor. Of the two video sequences, one is dedicated to training and the other to testing. They are taken under approximately the same illumination conditions, the same setup, almost the same background, and each face occupies between $1/4$ to $1/8$ of the image. This data base contains a variety of challenging operational conditions such as motion blur, out of focus factor, facial orientation, facial expression, occlusion, and low resolution. The number of ROIs detected varies from class to class, ranging from 40 to 190 for one video sequences.

Segmentation is performed using the well known Viola-Jones algorithm included in the OpenCV C/C++ computer vision library. In both cases, regions of interest (ROIs) produced are converted in gray scale and normalized to 24×24 images where the eyes are aligned horizontally, with a distance of

Algorithm 1 ADNPSO incremental learning strategy.

Inputs: An AMCS and new data sets D_t for learning.
Outputs: An ensemble of FAM networks.

Initialization:

- Set the swarm and archive parameters,
- Initialize all $(O+2)N$ networks,
- Set PSO iteration counter at $\tau = 0$, and
- Randomly initialize particles positions and velocities.

Upon reception of a new data block D_t, the following incremental process is initiated:

Update $FAM_{n,o}$ associated to the personal best positions:
for each particle n, where $1 \leq n \leq N$ **do**
  for each objectives o, where $1 \leq o \leq O$ **do**
    Train $FAM_{n,o}$ with validation.
    Estimate $f_o(\mathbf{h}^*_{n,o}, t)$.
Update the archive:
Update the accuracy of each solution in the archive.
Remove locally dominated solutions form the archive.
for each particle n, where $1 \leq n \leq N$ **do**
  for each objectives o, where $1 \leq o \leq O$ **do**
    Categorize $FAM_{n,o}$.
    if $FAM_{n,o}$ is a non-dominated solution for its network size domain **then**
      Add the solution to the archive.
      Remove solutions from the archive that are locally dominated by $FAM_{n,o}$.
Optimization process:
while stopping conditions are not reached **do**
  Update particle positions according to the ADNPSO algorithm (Equation 3).
  Estimate fitness and update personal best positions:
  for each particle n, where $1 \leq n \leq N$ **do**
    $FAM_n^{\text{est}} \leftarrow FAM_n^{\text{start}}$
    Train FAM_n^{est} with validation.
    Estimate $f_o(\mathbf{h}_n, t)$ of each objective.
    for each objective o, where $1 \leq o \leq O$ **do**
      if $f_o(\mathbf{h}_n, t) < f_o(\mathbf{h}^*_{n,o}, t)$ **then**
        $\{ \mathbf{h}^*_{n,o},\ FAM_{n,o},\ f_o(\mathbf{h}^*_{n,o}, t) \} \leftarrow \{ \mathbf{h}_n,\ FAM_n^{\text{est}},\ f_o(\mathbf{h}_n, t) \}$.
    Update the archive:
    Categorize FAM_n^{est}
    if FAM_n^{est} is locally non-dominated **then**
      Add the solution to the archive
      Remove solutions from the archive that are locally dominated by FAM_n^{est}.
  Increment iterations: $\tau = \tau + 1$.
Define initial conditions for D_{t+1}:
for each particle n, where $1 \leq n \leq N$ **do**
  $FAM_n^{\text{start}} \leftarrow FAM_n^{\text{est}}$.

12 pixels between them. Principal Component Analysis is then performed to reduce the number of features. The 64 features with the greatest eigenvalues are extracted and vectorized into $\mathbf{a} = \{a_1, a_2, ..., a_{64}\}$, where each feature $a_i \in [0,1]$ are normalized using the min-max technique. Learning is done with ROIs extracted from the first series of video sequences (1527 ROIs for all individuals) while testing is done with ROIs extracted from the second series of video sequences (1585 ROIs for all individuals).

Prior to simulations, each video data set is divided into blocks of data D_t, where $1 \leq t \leq T$, to emulate the availability of T successive blocks of training data to the AMCS. Supervised incremental learning is performed according to an *update* learning scenario. All classes are initially learned with the first block D_1; at a given time, face images of an individual becomes available and then learned by the AMCS to refined its internal models. In order to assess performance in cases where classes are initially ill defined, D_1 is composed of 10% of the data for each class, and each subsequent block D_t, where $2 \leq t \leq K+1$, is composed of the remaining 90% of one specific class.

The performance of the proposed ADNPSO learning strategy is evaluated using different optimization algorithms and different ensemble selection methods during supervised incremental learning of data blocks D_t. The parameters used are shown in Table 1. Weight values $\{w_1, w_2\}$ were defined as proposed in (Kennedy, 2007), and to detect a maximal number of local optima, no constraints were considered regarding the number of subswarm. Since Euclidean distances between particles are measured during optimization, the swarm evolves in a *normalized* $\mathbb{R}^4$ space to avoid any bias due to the domain of each hyperparameter. Before being applied to FAM, particle positions are de-normalized to fit the hyperparameters domain. For each new blocks of data D_t, the optimization is set to either stop after 10 iterations without improving the performance of either generalization error rate of network size, or after maximum 100 iterations. Learning is performed over 10 trials using ten-fold cross-validation. Incoming data is managed with the LTM as specified in (Connolly et al., 2012a). Each trial is repeated with five different class presentation orders, for a total of fifty replications.

Performance is evaluated for (1) an AMCS that uses the incremental learning strategy as described in Section 3: ADNPSO $\leftarrow$ the networks in the specialized archive corresponding to the phenotype local best plus a greedy search that maximizes genotype diversity (Connolly et al., 2012b). This system is compared to AMCSs using the ADNPSO learning strategy used with different optimization algorithms and ensemble

Table 1: ADNPSO parameters

Parameter	Value
Swarm's size N	60
Weights $\{w_1, w_2\}$	$\{0.73, 2.9\}$
Maximal size of each subswarm	4
Neighborhood size	5
Minimal distance between two local best	0.1
Minimal velocities of free particles	0.0001

selection techniques: (2) DNPSO ← the ensemble of FAM networks associated to the local best positions found with the mono-objective DNPSO (Nickabadi et al., 2008) plus a greedy search within the swarm to maximize genotype diversity (Connolly et al., 2012b), and (3) MOPSO ← the entire archive obtained with the ADNPSO incremental learning strategy employed with the multi-objective PSO algorithm that uses the notion of dominance to guide particles toward the Pareto optimal front (Coello et al., 2004). The MOPSO algorithm was used with the same applicable parameters than with the proposed ADNPSO, and with a grid size of 10 (for further details, see (Coello et al., 2004)).

The average performance of AMCSs is assessed in terms of generalization error and structural complexity. The *error rate* is the ratio of incorrect predictions over all test set predictions, where each face images is tested independently. Structural complexity is measure in terms of the *number of nodes on the F_2 layer* of all FAM networks in the ensemble.

5 RESULTS AND DISCUSSION

As depicted in Table 2, results indicate that using the proposed ADNPSO provides a level of accuracy that is comparable to that of using mono-objective optimization (DNPSO) but with a fraction of the computational cost. While the average network size of ensembles obtained with ADNPSO is the highest among all methods, the average ensemble size gives a total number of F_2 nodes that is lower than mono-objective optimization. On the other hand, while multi-objective PSO (MOPSO) yields the lightest ensembles, the error rate is on average 4% higher than that obtained with the other methods.

Table 3 shows performance with the average particle position and standard deviation after learning the whole data base. When the proposed ADNPSO directs subswarms of particles according information in the search spaces, rather than in the objective space, the swarm is able to remain dispersed in the objective space according both error rate and network size.

Table 2: Error rates and complexity indicators after incremental learning of data base. Complexity is evaluated in terms of ensemble size, average network F_2 layer size, and total F_2 layer size for the entire ensemble. Average values are presented with a 90% confidence interval.

Method	**ADNPSO**	DNPSO	MOPSO
Error rate (%)	22.4±0.6	22.7±0.7	26.9±0.7
Ensemble size	5.5 ± 0.4	12.4 ± 0.8	7.9 ± 0.6
Av. nb. of F_2 nodes	170±9	108±5	52±3
Tot. nb. of F_2 nodes	900±100	1300±100	420±50

Table 3: Average value and standard deviation of the swarm's current position after learning the entire IIT-NRC data base. Results are shown for the error rate and number of F_2 layer nodes, with a 90% confidence interval.

Method	**ADNPSO**	DNPSO	MOPSO
Average value			
Error rate(%)	35±2	19±1	45±2
Nb. of F_2 nodes	100±3	195±5	20±1
Standard deviation			
Error rate(%)	22±1	17±1	32±1
Nb. of F_2 nodes	110±5	90±4	10±1

The specialized archive insures that the most accurate solutions are stored for different network sizes.

On the other hand, with mono-objective optimization according only to accuracy (DNPSO), FAM networks tend to continuously grow their $F2$ layer to maintain or increase accuracy. The swarm then tends to have a higher level of accuracy with less variations, but with a much higher computational cost. Some particles are however still able to perform well with low structural complexities, explaining the relatively high dispersion for the number of F_2 nodes.

If influences are define in the objective space with the MOPSO algorithm, Table 3 shows that using classifiers such as FAM introduces a bias in the swarm's movements toward structural complexity. Theoretically, the MOO algorithms considers both objectives equally. However, given the nature of the problem (evolving FAM networks over time), a conventional MOO approach will find non-dominated solutions with fewer F_2 nodes more easily than non-dominated solutions with lower error rate. Particles in the different search spaces are then directed such as mostly minimizing FAM network size, thus limiting the search capabilities for accurate solutions.

6 CONCLUSIONS

This paper presents an incremental learning strategy based on ADNPSO that allows to evolve ensembles of heterogeneous classifiers in response to new reference data. This strategy is applied to an AMCS where all parameters of a swarm of FAM neural network classifiers (*i.e.*, a swarm of classifiers), each one corresponding to a particle, are co-optimized such that both error rate and network size are minimized. Multi-objective minimization is performed such that genotype diversity of solutions around local optima in the optimization search space, and phenotype diversity in the objective space are maintained. By using the specialized archive, local Pareto-optimal solutions detected by the ADNPSO algorithm can also be stored and combined with a greedy search algorithm to create ensembles based on accuracy, phenotype and genotype diversity.

Overall results indicates that using information in the search space of each objective (local optima positions and values), rather than in the objective space, permits creating pools of classifiers that are more accurate and with lower computational cost. For incremental learning scenarios with real-world video streams, ADNPSO provides accuracy comparable to that of using mono-objective optimization, yet requires a fraction of its computational cost. Since the proposed AMCS is designed with samples collected from changing classification environments, future work will focus on measures to detect various types of changes in the feature space (see Figure 1). This information could then be used to trigger an update of the pool and archive only when new data incorporated relevant information.

ACKNOWLEDGEMENTS

This research was supported by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

Brown, G., Wyatt, J., Harris, R., and Yao, X. (2005). Diversity creation methods: a survey and categorization. *Information Fusion*, 29(6):5–20.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., and Rosen, D. B. (1992). Fuzzy artmap: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Systems, Man, and Cybernetics C*, 3(5):698–713.

Coello, C., Pulido, G., and Lechuga, M. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279.

Connolly, J.-F., Granger, E., and Sabourin, R. (2008). Supervised incremental learning with the fuzzy artmap neural network. In *Proceedings of the Artificial Neural Networks in Pattern Recognition*, pages 66–77, Paris, France.

Connolly, J.-F., Granger, E., and Sabourin, R. (2012a). An adaptive classification system for video-based face recognition. *Information Sciences*, 192(1):50–70. Swarm Intelligence and Its Applications.

Connolly, J.-F., Granger, E., and Sabourin, R. (2012b). Evolution of heterogeneous ensembles through dynamic particle swarm optimization for video-based face recognition. *Pattern Recognition*, 45(7):2460–2477.

Elwell, R. and Polikar, R. (2011). Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531.

Gorodnichy, D. O. (2005). Video-based framework for face recognition in video. In *Second Workshop on Face Processing in Video in Proceedings of the Conference on Computer and Robot Vision*, pages 325–344, Victoria, Canada.

Granger, E., Henniges, P., Oliveira, L. S., and Sabourin, R. (2007). Supervised learning of fuzzy artmap neural networks through particle swarm optimization. *Journal of Pattern Recognition Research*, 2(1):27–60.

Jain, A. K., Ross, A., and Pankanti, S. (2006). Biometrics: A tool for information security. *IEEE Transactions on Information Forensics and Security*, 1(2):125–143.

Kennedy, J. (2007). Some issues and practices for particle swarms. In *Proceedings of the IEEE International on Swarm Intelligence*, pages 162–169, Honolulu, USA.

Kuncheva, L. I. (2004). Classifier ensembles for changing environments. In *Proceedings of the International Workshop on Multiple Classifier Systems*, pages 1–15, Cagliari, Italy.

Li, R., Mersch, T. R., Wen, O. X., Kaylani, A., and Anagnostopoulos, G. C. (2010). Multi-objective memetic evolution of art-based classifiers. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1–8, Barcelona, Spain.

Minku, L. L., White, A. P., and Yao, X. (2010). The impact of diversity on online ensemble learning in the presence of concept drift. *IEEE Transactions on Knowledge and Data Engineering*, 22(5):730–742.

Nickabadi, A., Ebadzadeh, M. M., and Safabakhsh, R. (2008). DNPSO: A dynamic niching particle swarm optimizer for multi-modal optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 26–32, Hong Kong, China.

Polikar, R., Udpa, L., Udpa, S. S., and Honavar, V. (2001). Learn++ : An incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man, and Cybernetics C*, 31(4):497–508.

Roli, F., Didaci, L., and Marcialis, G. (2008). Adaptive biometric systems that can improve with use. In Ratha, N. K. and Govindaraju, V., editors, *Advances in Biometrics*, pages 447–471. Springer London.

Valentini, G. (2003). *Ensemble methods based on bias-variance analysis*. PhD thesis, University of Genova, Genova, Switzerland.