2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication. September 9-13, 2012. Paris, France.

Incremental Learning using Partial Feedback for Gesture-based Human-Swarm Interaction

Jawad Nagi, Hung Ngo, Alessandro Giusti, Luca M. Gambardella, Jürgen Schmidhuber, Gianni A. Di Caro

Abstract— **In this paper we consider a human-swarm interaction scenario based on hand gestures. We study how the swarm can incrementally learn hand gestures through the interaction with a human instructor providing training gestures and correction feedback. The main contribution of the paper is a novel incremental machine learning approach that makes the robot swarm learn and recognize the gestures in a distributed and decentralized fashion using binary (i.e., yes/no) feedback. It exploits cooperative information exchange and swarm's intrinsic parallelism and redundancy. We perform extensive tests using real gesture images, showing that good classification accuracies are obtained even with rather few training samples and relatively small swarms. We also show the good scalability of the approach and its relatively low requirements in terms of communication overhead.**

I. INTRODUCTION

In this paper we study how a swarm of robots can *incrementally* and *cooperatively* learn hand gestures with a human instructor, who interactively provides training gestures and correction feedback. In particular, we consider the use of *partial (binary) feedback*. Once learned, the hand gestures can be visually recognized by the swarm and effectively used for human-swarm interaction tasks (e.g., to issue commands to the swarm, such as "go and search the environment").

In previous work [1], we considered the same gesture-based scenario for human-swarm interaction but using an *offline supervised learning* approach. We gathered a large dataset of gesture images and we used it to train a support vector machine (SVM) for gesture classification. During the interaction with the human, each robot uses its trained SVM to issue a probabilistic classification of the gesture, which can be more or less correct depending on the relative position of the robot with respect to the gesture and the presence of occlusions. The correctness of the classification performed by individual robots is also affected by the quality of the vision device and by the on-board available processing power, which are both relatively scarce for the robots typically used in a swarm To overcome these limitations, we developed a *distributed consensus* algorithm to let the robots fuse individual classifications and perform robust and accurate gesture classification at the swarm level.

While in this previous work we could achieve excellent swarm classification accuracies for different classes of gestures, the use of offline training is not entirely suitable for dynamic human-swarm interaction scenarios. In fact, it requires the prior definition of all signals (gestures) to classify and the gathering of a large training dataset. Moreover, the training dataset needs to be well representative of all the possible conditions of use of the system, which might be challenging when using low-quality vision devices (e.g., illumination conditions might significantly differ between training and normal usage, greatly affecting classification performance). Therefore, in this paper we build on our previous work, in particular on the notions of distributed sensing and consensus, but we consider online *incremental learning*, with a direct *interaction* between a human instructor and the swarm. In the dynamic scenario that we consider, training gestures are shown one at a time by a human instructor, when and where he/she wishes to do it. The swarm provides a response using flashing lights (encoding class labels) and in turn the human provides a correction feedback to the swarm, to let it learn incrementally and cooperatively.

In general *incremental learning tasks*, training examples are not available a priori, but come over time, usually one at a time, and the learning process might need to go on indefinitely. Moreover, the time between examples is not negligible in relationship to the way the learning agents make use of them, meaning that it is not feasible to wait gathering a large batch of samples before learning out of them. Therefore, incremental learning scenarios do not require the prior gathering of a large training set and can be used whenever it is required, in space and in time, making them more flexible and widely applicable than batch learning (e.g., see [2] and [3] for insightful discussions on incremental learning). In the context of robotics, incremental learning tasks are quite common: new experiences are typically gathered incrementally, and the environment changes over time, such that it is might not be always acceptable to wait before using the gathered examples. This is also the case of our human-swarm interaction scenario, where incremental learning can allow the human to teach new gestures, or improve what the swarm has learned so far, by providing new training gestures on the spot, under the desired external conditions (e.g., regarding illumination). Moreover, since the robots used in swarms usually have limited sensing and processing capabilities, resulting in noisy input data, the general ability of remembering past observations and incrementally update the related models is extremely useful [4].

While the use of incremental learning through direct interaction fits our scenario, in practice, it might not be feasible to ask a human instructor to go through a large number of learning interactions with the swarm. Therefore, in this paper we also address the issue of how a swarm can effectively learn *after a few teaching interactions*. At this

All authors are with the Dalle Molle Institute for Artificial Intelligence (IDSIA), Manno–Lugano, Switzerland. {jawad,hung, alessandrog,luca,juergen,gianni}@idsia.ch

aim, we leverage on the intrinsic *parallelism* and *redundancy* of a swarm. On the one hand, we exploit the fact that each robot sees the gesture from a different point of view. On the other hand, we let the robots *cooperate*, by sharing among each other their training data through multi-hop wireless communications. In this way, also for the case of relatively small swarm sizes, we can make the swarm rapidly and effectively learn after a few interaction rounds with the human (shown in the experimental results of Section VI).

Since we make use of an interactive incremental learning approach, we need to define how the human instructor provides *correction feedback*. For multi-class interactive learning tasks, such as the classification of multiple hand gestures, feedback may be typically represented as either *full* or *partial*. In the full feedback approach [5], [6] the actual label of the observation (gesture class) is directly provided as feedback to update the learner model. Partial, and in particular, *binary* feedback relates to the predicted outcome as being *true* (yes) or *false* (no), without providing an explicit label for the given input. Compared to full feedback, the use of binary feedback is more flexible and comes at a lower cost: it is not needed to create a full set of labelled samples and/or define ways to communicate the labels. This is particularly important in dynamic interactive scenarios like ours: it might be problematic for the instructor to precisely remember the label of each gesture and provide it to the robots only using gestures. Good memory, attention, and additional devices (other than her/his hands) would be required (e.g., a computer system with a wireless interface). On the other hand, it is straightforward and less error-prone to provide a yes/no feedback (e.g., waving the hands).

Therefore, based on these insights, in this paper we present a novel algorithm for single and multi-agent multi-class incremental learning using binary feedback. The algorithm is derived from an additional algorithm that we also propose for the full feedback case. Both algorithms are based on extensions of the *Regularized Least Squares* (RLS) algorithm. In the experiments described in Section VI we equip the robots in the swarm (using the *foot-bot* robots [7]) with the algorithms and we show that, once combined with the information sharing mechanisms for cooperative learning and consensus, both algorithms allow the swarm to achieve good gesture recognition accuracy and scalability (the more robots, the better the performance). In particular, the difference between the performance of the two algorithms is not significantly large, making viable and effective in our scenario to use binary feedback, which is intrinsically more flexible and easy to use for a human instructor. In the experiments we also study the impact of different communication strategies, showing that the algorithms can achieve good accuracies even with a limited communication overhead.

The rest of the paper is organized as follows. Section II discusses related work for incremental learning. Section III presents the two algorithms for interactive incremental learning. Section IV briefly describes the approach used by an individual robot to process, interactively learn and recognize gesture images. How a swarm of robots performs the same tasks in cooperation is described in Section V. The experimental results for learning and recognition performance under different conditions for number of robots and information sharing are reported in Section VI. Conclusions and future work are outlined in Section VII.

II. RELATED WORK

In order to constantly learn new information from dynamic environments, a number of studies have focused on developing online variants of supervised machine learning algorithms such as, multi-layer perceptrons (i.e., Neural Networks) and kernel-based methods (i.e., Support Vector Machines–SVMs). Recently, a few authors have proposed the use of partial feedback, which build upon results from natural reasoning (i.e., similar to teaching and reasoning with infants) while coming at a lower cost as compared to full feedback systems. Prominent work on the use of partial feedback [8], [9] is based on scalable active learning with active selection from a pool of unlabelled samples. However, since no training set is available a priori in interactive learning, this approach is not directly applicable to the scenario we consider in this paper.

In the context of partial feedback, many researchers in the machine learning community have proposed supervised learning algorithms with the capability of performing updates using partial feedback, referred to as the class of *contextual bandit* algorithms. Initial work on bandit algorithms [10] for online multi-class prediction provided the first step towards supervised learning using partial feedback. In this context, an improved Banditron was proposed [11] in which the multi-class prediction problem is reduced to a binary problem, based on a conservative one-versus-all reduction scheme.

An exponential gradient descent algorithm with partial feedback for multi-class online learning was introduced in [12], where the mistake bound of the algorithm is independent from the dimension of data, thereby, making the approach suitable for classifying high dimensional data and outperforming the Banditron. Since the Banditron maximizes the run-time gain by balancing between the exploration and exploitation with a fixed trade-off parameter, a few studies have considered defining this trade-off more effectively. In this context, learning strategies to automatically adjust the trade-off parameter for the Bandit model were presented [13]. In [14], a multi-class classification scheme was introduced with the bandit feedback using adaptive regularization. The approach is based on the 2^{nd}-order perceptron and makes use of the *upper confidence bounds* (UCBs) [15] to balance the trade-off between exploration and exploitation for improving the bounds of the Banditron [10]. Based on these results, we also include UCBs, in a modified form, in our algorithm.

All the algorithms described so far based on the Banditron and its extensions require tuning a number of parameters and a large number of samples for producing good recognition results and gaining confidence (see Figure 5). On the other hand RLS [16] methods based on linear regression has shown performance better than the Banditron class of algorithms [14] and similar to that of SVMs and other kernel-based methods [16], with the additional ability to provide confidence estimates of the predicted samples. Therefore, in this paper we propose interactive incremental learning algorithms derived from RLS methods.

III. INTERACTIVE LEARNING USING FEEDBACK

The interactive learning algorithms presented in this Section are inspired by the work in [17] and [18]. During the interactive learning process, at each interaction step t, the learner observes a length-d input vector $\mathbf{x}_t \in \mathbb{R}^d$ and predicts the label $\hat{y}_t \in \{1, ..., M\}$ among M given classes. The learner observes the outcome related to the prediction it has issued, which can be the true class label $y_t \in \{1, ..., M\}$ (in the full feedback setting) or a partial (binary) signal $l_t \in \{-1, 1\}$ indicating whether the prediction was correct or not. Exploiting the feedback information, the learner updates its learning parameters with the objective of minimizing the relative loss compared to the best parameters in the given model [17], [18].

In our scenario we consider learning from a linear model. More specifically, our algorithms are extensions of the *online ridge regression*, a linear regression model predicting the output $\gamma_t \in \mathbb{R}$ by $\gamma_t = \mathbf{w}^\top \mathbf{x}_t$, where $\mathbf{w} \in \mathbb{R}^d$ represents a weight vector. Given a set of T training instances $(\mathbf{x}_1, \dots, \mathbf{x}_T)$, and its associated set of outputs $(z_1, \dots, z_T)$, ridge regression aims to minimize the sum of regularized square errors, thus, named Regularized Least Squares methods [16]:

$$L_{Ridge}(\mathbf{w}) = \sum_{i=1}^{T} (z_t - \mathbf{w}^\top \mathbf{x}_t)^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2 \tag{1}$$

where λ is a regularization parameter controlling the trade-off between the goodness-of-fit and the simplicity of the weights. Taking the derivatives and equating them to zero gives the closed-form solution of $\mathbf{w}$:

$$\mathbf{w} = (\lambda I + \sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t^\top)^{-1} (\sum_{t=1}^{T} z_t \mathbf{x}_t) \tag{2}$$

For notation convenience, we denote:

$$A = (\lambda I + \sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t^\top)^{-1}, \tag{3}$$

$$\mathbf{b} = \sum_{t=1}^{T} z_t \mathbf{x}_t^\top. \tag{4}$$

In the next section we describe our modified RLS algorithm that supports multi-class online predictions with full feedback. From this algorithm we derive in Section III-B the algorithm that deals with partial (binary) feedback.

A. Full Feedback

We use the one-vs-all approach [19] to create M RLS predictors sharing the same data correlation matrix A but having different row vectors $\mathbf{b}^i$. The procedure is described in Algorithm 1. Following the "forward" method as described in [17], [18], we make a small but important modification to the original RLS algorithm, to update matrix A using the current instance $\mathbf{x}_t$ *before* making predictions (line 5 and 6 in Algorithm 1). This approach improves the relative loss bound of the online learning algorithm, as it has been shown in [17], [18]. Our tests (not reported here) also confirmed an improvement in the cumulative loss.

After the true label y_t is observed, for each RLS predictor the corresponding binary label $z_t^i \in \{-1, +1\}$ is inferred and used for model updating. The RLS margins γ_t^i indicate the degree of belief of predictor i when predicting that the current instance $\mathbf{x}_t$ belongs to class i. The final prediction outcome is thus decided by the most confident label (that with the highest prediction margin). To have a probabilistic interpretation of the predictions over all M classes, we can project the margins $\boldsymbol{\gamma}_t = (\gamma_t^1, ..., \gamma_t^M)^\top$ onto a prediction simplex. Efficient algorithms for solving this minimization problem are available, e.g., [20].

Algorithm 1: RLS-based multi-class online prediction with full feedback (parameters: λ, M, d)

```
   //Initialization
1  $A \leftarrow \frac{1}{\lambda}\mathbf{I}$ // $d \times d$ matrix
2  for $i = 1 : M$ do $\mathbf{b}^i \leftarrow \mathbf{0}$ // $1 \times d$ vector
   //Main interactive learning loop
3  for $t = 1, 2, ...$ do
4      Observe new input $\mathbf{x}_t \in \mathbb{R}^d$
       //Incorporate current input $\mathbf{x}_t$ using
         Sherman-Morrison incremental update
5      $A \leftarrow A - \frac{(A\mathbf{x}_t)(A\mathbf{x}_t)^\top}{1+\mathbf{x}_t^\top A \mathbf{x}_t}$
       //Each predictor issues a prediction
6      for $i = 1 : M$ do $\gamma_t^i \leftarrow \mathbf{b}_{t-1}^i A \mathbf{x}_t$
       //Select predictor with highest margin
7      Predict $\hat{y}_t \leftarrow \mathrm{argmax}\gamma_t^i$
8      Observe true label $y_t \in \{1, ..., M\}$
       //Update each model accordingly
9      for $i = 1 : M$ do $\mathbf{b}^i \leftarrow \begin{cases} \mathbf{b}^i + \mathbf{x}_t^\top, & y_t = i \\ \mathbf{b}^i - \mathbf{x}_t^\top, & y_t \neq i \end{cases}$
10 end
```

B. Partial Feedback

When the binary feedback is a "correct prediction" (i.e., $l_t = 1$), we can easily infer the corresponding binary label $z_t^i \in \{-1, +1\}$ for each RLS predictor, just as the case with full feedback. However, when the feedback is an "incorrect prediction" (i.e., $l_t = -1$), we can only infer just the binary label z_t^i of the chosen predictor i. Thus, the update rule is modified accordingly, as described in Algorithm 2 (lines 14–22). Since each predictor may be updated differently, besides separate vectors $\mathbf{b}_i$ as in the full feedback case, we need to maintain a separate matrix A_i for each predictor i.

It has been proved (e.g., see [21], [22], [17]) that the variance in estimating the RLS margin γ_t^i on instance $\mathbf{x}_t$

can be upper bounded by the quantity $\mathbf{x}_t^\top A^i \mathbf{x}_t$. Hence, with high probability, the Bayes optimal decision margin β_t^i will be in the range:

$$\gamma_t^i - \sigma_t^i \leq \beta_t^i \leq \gamma_t^i + \sigma_t^i, \tag{5}$$

where $\sigma_t^i := \sqrt{\mathbf{x}_t^\top A^i \mathbf{x}_t}$ represents such confidence interval (CI). We adopt the widely used upper confidence bound UCB $= \gamma_t^i + \sigma_t^i$ [15], [14] to optimistically select the predictor (line 12 of Algorithm 2).

Algorithm 2: RLS-based multi-class online prediction with binary feedback (parameters: λ, M, d)

```
   //Initialization
1  for $i = 1 : M$ do
2      $A^i \leftarrow \frac{1}{\lambda}\mathbf{I}$ // $d \times d$ matrix
3      $\mathbf{b}^i \leftarrow \mathbf{0}$ // $1 \times d$ vector
4  end
   //Main interactive learning loop
5  for $t = 1, 2, ...$ do
6      Observe new input $\mathbf{x}_t \in \mathbb{R}^d$
7      for $i = 1 : M$ do
           //Incorporate current input $\mathbf{x}_t$
8          $B^i \leftarrow A^i - \frac{(A^i\mathbf{x}_t)(A^i\mathbf{x}_t)^\top}{1+\mathbf{x}_t^\top A^i \mathbf{x}_t}$ ;
           //Predict with confidence interval
9          Margin: $\gamma_t^i \leftarrow \mathbf{b}^i B^i \mathbf{x}_t$
10         CI: $\sigma_t^i \leftarrow \sqrt{\mathbf{x}_t^\top B^i \mathbf{x}_t}$
11     end
       //Select predictor with highest UCB
12     Predict $\hat{y}_t \leftarrow \mathrm{argmax}(\gamma_t^i + \sigma_t^i)$
13     Observe binary feedback $l_t \in \{-1, 1\}$
14     if $l_t = 1$ then
           //Update each predictor accordingly
15         for $i = 1 : M$ do
16             $A^i \leftarrow B^i$
17             $\mathbf{b}^i \leftarrow \begin{cases} \mathbf{b}^i + \mathbf{x}_t^\top, & \hat{y}_t = i \\ \mathbf{b}^i - \mathbf{x}_t^\top, & \hat{y}_t \neq i \end{cases}$
18         end
19     else
           //Update only selected predictor
20         $A^{\hat{y}_t} \leftarrow B^{\hat{y}_t}$
21         $\mathbf{b}^{\hat{y}_t} \leftarrow \mathbf{b}^{\hat{y}_t} - \mathbf{x}_t^\top$
22     end
23 end
```

It is worth mentioning that, similar to the approach employed in [14], our Algorithm 2 also maintains additional confidence information associated with each prediction, making an optimistic prediction using the highest UCB. However, as opposed to the algorithm in [14], our algorithm maintains separate binary predictors in a one-vs-all approach, allowing all the predictors to get updated when a positive feedback is available (lines 14–18 in Algorithm 2), being in this way more label-efficient. Furthermore, the algorithm in [14], which is based on the original RLS, does not update matrix A using the current instance $\mathbf{x}_t$ *before* making predictions.

IV. SINGLE-ROBOT INTERACTIVE LEARNING

In this section we describe how the algorithms are employed in our hand gesture recognition for the case of a single robot. The swarm case is described in Section V.

The recognition of hand gestures starts with the robot detecting a human performing a hand gesture by using its on-board camera. The robot acquires an image of the hand gesture, processes the acquired image in order to extract relevant information regarding the gesture, and then performs a classification of the gesture based on a predefined set of possible known gestures. The output of this process is a *posterior probability* vector assigning a confidence to each predefined gesture class. The process is iterated through the repeated acquisition of new images for different gestures as they are provided by the human.

To simplify the recognition task without losing generality, we assume that the human wears a glove with a known characteristic color. We consider a set of $M = 6$ gestures in terms of *finger counts* (as illustrated in Section VI), in which zero (i.e., the closed hand) to five fingers are shown as illustrated in Figure 1. To make the problem more challenging, gestures are shown in any rotation of the hand and in any finger combinations for representing finger counts.

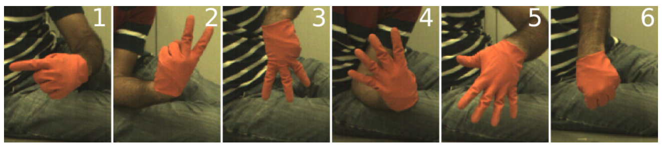

Fig. 1: The six hand-gesture finger counts.

To deal with the recognition task, we adopt a basic machine vision approach based on *segmentation*, *supervised feature extraction*, and *supervised classification*–inclusive of feedback. We accept that a single robot may have poor recognition performance, especially when observing the hand from bad POVs. Our main goal is in fact to obtain good classification performance at the swarm-level, by fusing posterior probabilities from multiple robots (some of which will hopefully be in good positions), as discussed in Section V.

A. Color-based Segmentation

Since the hand projection (in the acquired RGB image by the camera sensor of the robot) normally covers a very small fraction of the acquired RGB image, we segment the hand gesture by exploiting the characteristic color (orange in our case) of the glove, using a standard color-based segmentation approach in the HSV (Hue, Saturation, Value) color space.

After color-based segmentation, black-and-white (i.e, *binary* $[0, 1]$ pixel) images are generated, where pixels corresponding to 0 represent the background and pixels with the intensity value 1 represent the segmented hand gesture (i.e., the orange coloured glove) as shown in Figure 2. Next, *connected component analysis* is performed on the resulting binary images and the connected component (i.e., the blob) of *maximal area* is identified as the observed hand gesture.

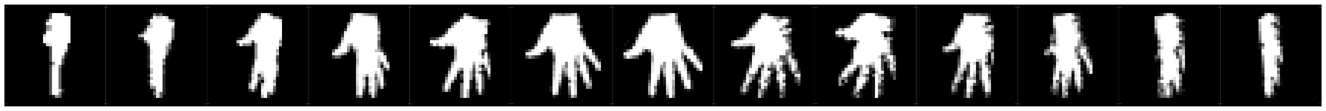
Fig. 2: Segmentation results for images of the same gesture acquired from different points of view (POV).

B. Supervised Feature Extraction

Using the segmented images of the blob, a polygon is fitted to the blob contour to smooth noisy and rough edges in the silhouette of the hand gesture. Next, using the fitted contour of the polygon, numerical features are computed using a total of $N_{features} = 110$ *geometrical properties* (e.g. shape and blob characteristics, image moments etc.) frequently used in the literature on similar tasks (see [1], [23] for details).

We determine the weight of the features towards the classification problem by computing the *Information Gain* of the each of the 110 features. Therefore, the complete set of $N_{features} = 110$ features computed from a binary image i corresponds to a feature descriptor $\mathbf{f}_i$.

C. Supervised Classification with Feedback

After a feature descriptor $\mathbf{f}_i$ is computed from a sensed image, the incremental learning algorithms with feedback of Section III are used to classify the observation, resulting in a posterior probability vector $\mathbf{p}$ of M elements. The class corresponding to the largest value in $\mathbf{p}$ is the outcome (i.e., the recognized gesture class) determined by the robot.

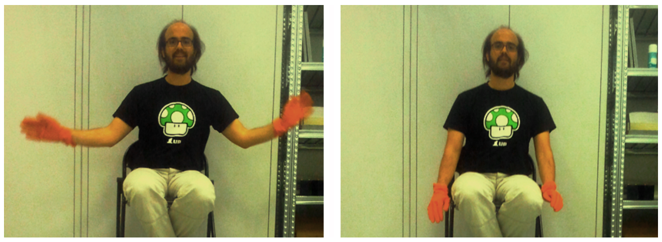
Fig. 3: Vision-based motion detection used for providing partial feedback for: *incorrect prediction* (left) and *correct prediction* (right).

After the given gesture is classified, the human instructor provides a feedback, based on the predicted outcome of the gesture. For providing binary feedback to use with Algorithm 2, we employ a vision-based motion detector, where the human waves their arms indicating that the feedback for the predicted outcome is *false* (incorrect), while if no motion is detected (the human does not move the arms) then the feedback for the predicted outcome is *true* (correct), as illustrated in Figure 3. In the case of full feedback, the actual label (i.e., the gesture class) of the predicted observation is provided to the robot by means of an appropriate wireless-based interface. After the feedback has been assessed by the robot, the robot incrementally updates its classifier with the predicted observation as prescribed by the algorithms.

V. MULTI-ROBOT INTERACTIVE LEARNING

As pointed out in the Introduction, the recognition performance of a single robot can be quite poor, being strongly affected by its position with respect to the gesture. Also its learning curve might be quite flat, since it can only rely on the training samples it incrementally gathers. When a swarm of robots is used, the limitations of individual robot performance can be overcome by letting the available set of robots cooperate during both learning and recognition.

In the following, we let $\{r_1, r_2, \ldots, r_{N_{robots}}\}$ to represent the set of robots in a swarm of size N_{robots}. When the swarm is idle or is performing some routine activity, a subset of the robots is continuously scanning the environment to detect the presence of a human agent and his/her willingness to communicate. This is indicated by the presence of an orange-coloured glove. When a robot detects it in its field of view, it locally broadcasts a special start message which is then relayed to the rest of the swarm in a multi-hop fashion using a wireless interface.

A. Sensing and Communication

As described in Section IV, each robot r in the swarm iteratively acquires and processes gesture images. After the acquisition of an image i, a robot r computes the feature descriptor $\mathbf{f}_i^r$ and performs the following actions for classification and learning:

- *Classification*: The feature descriptor $\mathbf{f}_i^r$ is used to to classify the presented gesture according to the local $r_{classifier}$, which results in a *posterior probability* vector $\mathbf{p}_i^r$, which is propagated to the rest of the swarm encoded in a `OPINION` message. Since in our algorithm γ is the actual prediction margin, for each binary predictor margin, γ_i is projected onto a simplex to obtain $\mathbf{p}_i^r$. The `OPINION` messages are used for *consensus-building* at swarm level (i.e., making the swarm reach an agreement upon the presented gesture).
- *Learning*: The feature descriptor $\mathbf{f}_i^r$ is locally broadcast and propagated in multi-hop modality to the rest of the swarm encoded in a `DESCRIPTOR` message, which is used to perform *cooperative learning* within the swarm. When a robot s first receives the feature descriptor $\mathbf{f}_i^r$ sent out by robot r, it uses it to incrementally update its local classifier $\mathbf{s}_{classifier}$ as described in Section III.

B. Swarm-level Decision-making

For robot r and image i, the posterior probability vector $\mathbf{p}_i^r$ represents the *opinion* of r about the hand gesture, and the element in the $\mathbf{p}_i^r$ with the largest value is the predicted outcome (i.e., the recognized gesture class). In our algorithms in Section III, the posterior probability vectors are conveniently normalized, so their sum equals to 1 over all classes, which is expressed by $\sum_{i=1}^{M} \mathbf{p}_i^r = 1$. In order to obtain a swarm-level prediction of the hand gesture which accounts for all the different existing opinions, some form of *distributed consensus* is needed. We adopt a quite straightforward approach to implement this: For each given gesture i, at each robot we fuse the $\mathbf{p}_i^r$ obtained from all robots $r = 1, \ldots, N_{robots}$ from their different POVs using a simple averaging approach: $\delta^i = \sum_{r=1}^{N_{robots}} \mathbf{p}_i^r$, where δ^i represents the mean of the posterior probability vectors (i.e., the outcome of the distributed consensus).

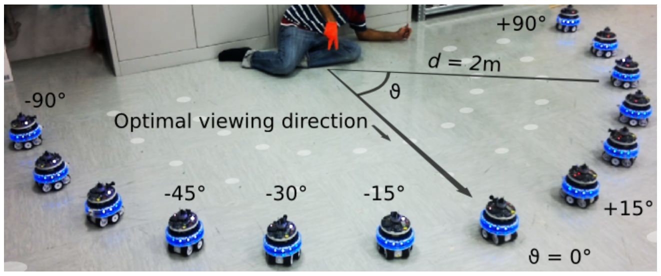

Fig. 4: The experimental setup of the robot swarm for data acquisition.

Operatively, information fusion is realized through the multi-hop spreading of the `OPINION` messages described in Section V-A. On reception of an `OPINION` message, each robot r computes its local δ^r. As more `OPINION` messages are received the local value of δ^r is iteratively updated. The process stops after each robot has received $N_{robots} - 1$ different `OPINION` messages implying that the value of δ computed locally by all robots is the same, and the largest value in δ is determined as the predicted outcome. In this state, the swarm has *converged* to a unanimous decision, accounting for all the possibly different opinions of robots in the swarm. Finally, all robots flash their lights to communicate the swarm predicted outcome to the human.

In our previous work [1] we developed a more sophisticated strategy for distributed sensing and consensus, leveraging on robot mobility, multi-hop communications, and statistical reasoning. In this paper, we opted for a simpler strategy to let the contribution of the incremental learning algorithms to stand out in a clearer way.

C. Cooperative Learning

Cooperative learning, which amounts to *collaborative training*, follows a process similar to that of cooperative decision-making. `DESCRIPTOR` messages are spread throughout the swarm according to the same multi-hop mechanisms employed for `OPINION` messages (in practice, flooding with a control on message relaying in order to minimize the number of duplicate packet messages). Once a robot has received all the N_{robots} descriptors for the current image, it uses this information, including its own local descriptors, in the algorithms presented in Section III to perform incremental learning (i.e., to update its local classifier). The process iterates as new gestures are presented.

VI. EXPERIMENTAL RESULTS

We employed a swarm of 13 foot-bots [7] (see Figure 4) to acquire hand gesture images from different POVs with known *ground truth*. We gathered a dataset of $74,000$ images from 65 different view points. [1] Using these data, we performed extensive *emulation tests* (i.e., using real images) to assess the learning performance of Algorithms 1 and 2.

In the experiments, the robots in the swarm are initially deployed in random positions sampled from the positions from where the real images in the dataset were acquired. Robots exchange messages according to a low-bandwidth (100 bytes/sec) infra-red (IR) based line-of-sight wireless device. Initially, each robot trains its individual classifier (based on the Algorithms in Section III) using $H_{initial}$ samples from each of the $M = 6$ gesture classes, such that $T = H_{initial} \times M$ is the total number of samples used for the *initial training phase* of the swarm. This phase is used to start-up the learning process, keeping $H_{initial}$ limited a few training samples. After the initial training phase, the swarm and the human instructor go through a series of *interaction rounds*. At the beginning of each round, the robots move (e.g., to simulate the fact that the swarm performs an assigned task). This is implemented by randomly changing robot positions, always based on the 65 different view points in the dataset. During the interaction round, robots do not move, and $i = 1, \ldots, H_{interact}$ gestures are shown to the swarm. The class $k \in \{1, ..., M\}$ of each gesture is randomly selected from the 6 finger count classes. For each gesture i of class k, each robot "sees" an image of class k sampled from the subset of dataset images of class k which were acquired from the robot's current position.

The swarm performs a distributed consensus $H_{interact}$ times, generating a set of δ values: $\delta_{interact} = \{\delta_i | i = 1, \ldots, H_{interact}\}$. The largest element in δ_i for each gesture represents the predicted outcome of the swarm. Comparing these values with the expected outcomes for gesture classification, for each interaction round we compute how many gestures were identified correctly by the swarm ($H_{correct}$) out of the total $H_{interact}$ samples predicted: $\frac{H_{correct}}{H_{interact}}$.

Based on the outcome of each δ_i, the human-instructor provides a feedback to the swarm, that assigns the actual (full) or true/false (binary) label y_i for each of the predicted $H_{interact}$ samples. This feedback information is used to incrementally update the interactive learning algorithms. Then the robots move, changing again their positions relative to the hand gesture, and the next round starts. If the robots did not move, after a few training gestures, they could learn very effectively to classify gestures from their specific positions. On the other hand, we want to put to test the ability of the algorithms to generalize over different view points exploiting the presence of a distributed swarm.

The first set of tests considers the *convergence rate*, in terms of classification accuracy, of Algorithms 1 and 2.

[1] A video with the description of the experimental procedure, as well as of the real-time performance of the system in the context of our previous work using offline training [1], is available here: `http://www.idsia.ch/~gianni/SwarmRobotics/swarm_robotics.html#gestures`

Results are shown in Figure 5, using $\lambda = 1$. Results for the *Banditron* algorithm [10] according to the implementation in [14], are hereafter also included in order to compare the performance of our algorithms vs. the state-of-the-art. The setting of the experiment is as follows: $N_{robots} = 13$, $N_{features} = 110$, $H_{initial} = 10$, $H_{interact} = 10$. It can be observed that the swarm recognition accuracy for the partial feedback approach closely follows the full feedback approach, while coming at a lower cost. Both algorithms achieve good classification accuracies after a few interaction rounds. The algorithm based on full feedback is able to achieve 100% accuracy after 15 rounds, while the use of partial feedback seems to reach a plateau around 90%. The Banditron, which uses randomized updates, requires a far larger number of samples to converge as compared to our algorithms.

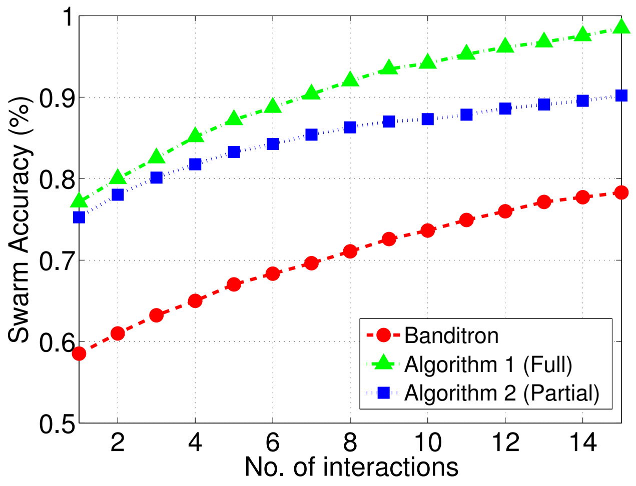

Fig. 5: Convergence rate of the proposed interactive learning approaches in Algorithm 1 and 2 with the Banditron [10].

The results in Figure 6 show the *scalability* of performance: the effect of the size of the robot swarm (from 1 to 30 robots) on the classification accuracy of the algorithm using binary feedback. The experimental setting is as follows: $N_{features} = 110$, $H_{initial} = 20$, $H_{interact} = 15$. A good, near-linear proportionality between performance and interaction rounds can be observed for all the considered swarm sizes. Moreover, as expected, using more robots results in better and faster learning. A saturation in performance seems to be achieved at 30 robots: adding more robots would not result in significant performance improvement.

In Figure 7 we show the effect of the number of features computed from the binary (segmented) images (i.e., used for learning and recognition) on the overall recognition rate of the swarm when using the binary feedback algorithm. The experimental parameters in this case are: $N_{robots} = 13$, $H_{initial} = 15$, $H_{interact} = 10$. As expected, the number of features used for learning and classification have a significant impact on swarm-level recognition. From the full set of $N_{features} = 110$ selected features, we can see that a subset of $N_{features} = 40$ features having the largest *information*

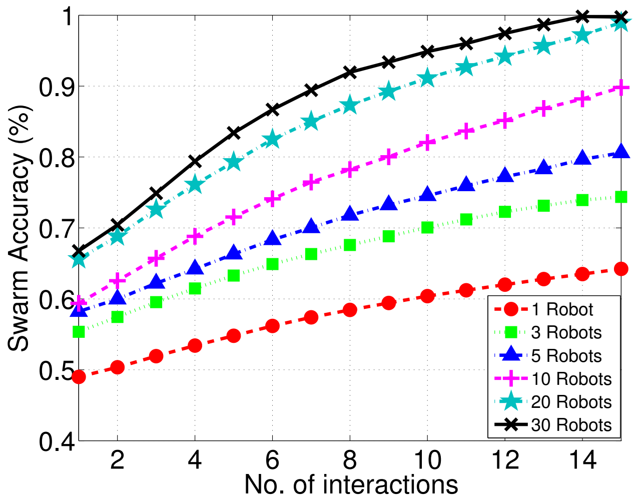

Fig. 6: The effect of the number of robots on the classification rate of Algorithm 2, based on binary feedback.

gain represents the best trade-off. This indicates that due to noise and ambiguities in the environment, not all the 110 features that we computed from the binary images provide a good discrimination in the feature space for our dataset. Performing dimensionality reduction (i.e., select a subset of features with high discrimination power) has a twofold advantage in our case: (i) it improves the speed of learning and classification, and (ii) reduces the communication overhead by reducing the payload of the exchanged messages.

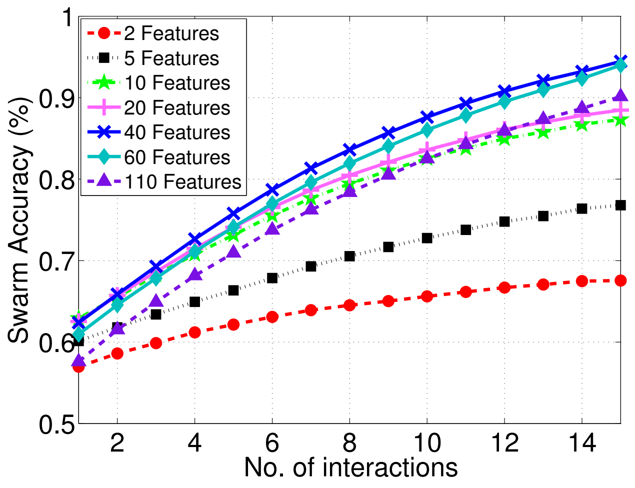

Fig. 7: The impact of different number of features on the classification rate of Algorithm 2.

The results in Figure 8 show the recognition performance of the swarm with different *communication strategies* for Algorithm 2. The parameters of the experiments are: $N_{robots} = 13$, $N_{features} = 40$, $H_{initial} = 2$. In the 1-hop communication scenario, after one hop, messages are no further relayed in the wireless robot network. In the n-hops case, messages are relayed up to n hops. In full communication the robot network is fully connected, such that each robot message reaches all other robots. In the case of no communication, robots do not exchange messages. These different models implement different ways of mutual information spreading throughout the swarm and have a different impact in terms of communication overhead. From the figure we can observe that, in general, more communication implies more cooperation, and results in better classification performance. However, for the considered swarm size, 2-hops communication represents the best trade-off between swarm accuracy and communication overhead: the use of n-hops, with $n > 2$ would not significantly improve accuracy but would result in more communication overhead.

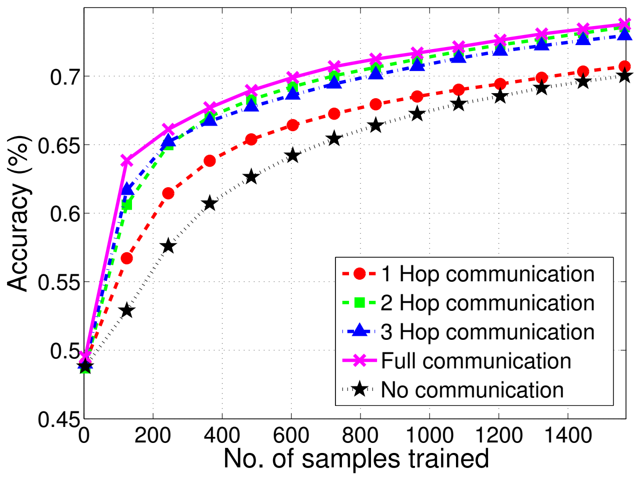

Fig. 8: The effect of using different communication strategies on the classification rate of the swarm.

VII. Conclusions

We considered a human-swarm interaction scenario based on hand gestures. The human plays the role of an instructor who interactively and incrementally provides training gestures and correction feedback. We focused on the use of partial (i.e., binary, yes/no) feedback, which is easy to provide through gestures and less error-prone compared to full feedback. We presented a novel algorithm for single- and multi-agent multi-class incremental learning based on binary feedback, as well as an algorithm for full feedback. Both algorithms are based on regularized least squares methods. We equipped the robots in the swarm with the algorithms and, by exploiting information sharing, cooperation, and distributed consensus within the swarm, we studied their effectiveness for learning and recognition of hand gestures. In a number of tests based on real images gathered using foot-bot robots, we have shown that, in terms of recognition accuracy, both algorithms have good convergence properties and scale well with respect to swarm size. In particular, we have shown the good overall performance that can be achieved just using binary feedback through swarm cooperation and with minimal requirements in terms of communication overhead.

Future work will consider validation using larger sets of gesture classes and the use of the approach in real-world human-swarm interaction and cooperation scenarios.

Acknowledgments

This research was supported by the Swiss National Science Foundation (SNSF) through the National Centre of Competence in Research (NCCR) Robotics, the 7th Framework Program of the EU under grants #231722 (IM-Clever project), and by the SNSF grant CRSIKO-122697 (Sinergia project).

References

[1] A. Giusti, J. Nagi, L. Gambardella, and G. A. Di Caro, "Cooperative sensing and recognition by a swarm of mobile robots," in *Proceedings of the 25th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012 (to be published).

[2] C. Giraud-Carrier, "A note on the utility of incremental learning," *AI Commununications*, vol. 13, no. 4, pp. 215–223, 2000.

[3] W.-M. Shen, "Efficient incremental induction of decision lists – can incremental learning outperform non-incremental learning?" University of Southern California, Tech. Rep. USC-ISI-96-012, 1996.

[4] B. D. Argall, S. Chernova, *et al.*, "A survey of robot learning from demonstration," *Rob. & Aut. Sys.*, vol. 57, no. 5, pp. 469–483, 2009.

[5] G. Pirlo, C. A. Trullo, and D. Impedovo, "A feedback-based multi-classifier system," in *Proc. of ICDAR*, 2009, pp. 713–717.

[6] D. Impedovo and G. Pirlo, "Updating knowledge in feedback-based multi-classifier systems," in *Proc. of ICDAR*, 2011, pp. 227–231.

[7] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. L. Christensen, A. Decugnière, G. A. Di Caro, and other 22 authors, "Swarmanoid: a novel concept for the study of heterogeneous robotic swarms," *IEEE Robotics & Automation Magazine*, 2012 (to appear).

[8] A. J. Joshi, F. Porikli, *et al.*, "Breaking the interactive bottleneck in multi-class classification with active selection and binary feedback." in *Proc. of the IEEE Intl. Conf. on CVPR*, 2010, pp. 2995–3002.

[9] ——, "Scalable active learning for multi-class image classification," *IEEE Trans. on PAMI*, vol. (to appear), 2012.

[10] S. M. Kakade *et al.*, "Efficient bandit algorithms for online multiclass prediction," in *Proc. of the IEEE ICML*, 2008, pp. 440–447.

[11] G. Chen, G. Chen, *et al.*, "Beyond banditron: A conservative and efficient reduction for online multiclass prediction with bandit setting model," in *Proc. of the 9th IEEE ICDM*, 2009, pp. 71–80.

[12] S. Wang, R. Jin, and H. Valizadegan, "A potential-based framework for online multi-class learning with partial feedback," *J. of Machine Learning Research*, vol. 9, pp. 900–907, 2010.

[13] H. Valizadegan, R. Jin, and S. Wang, "Learning to trade off between exploration and exploitation in multiclass bandit prediction," in *Proc. of the 17th ACM Intl. Conf. on KDD*, 2011, pp. 204–212.

[14] K. Crammer *et al.*, "Multiclass classification with bandit feedback using adaptive regularization," in *Proc. of ICML*, 2011, pp. 273–280.

[15] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs," *J. of Machine Learning Research*, vol. 3, pp. 397–422, 2003.

[16] R. Rifkin, G. Yeo, and T. Poggio, "Regularized least squares classification," *Advances in Learning Theory: Methods, Models and Applications NATO Science Series III*, vol. 190, pp. 131–154, 2003.

[17] K. S. Azoury and M. K. Warmuth, "Relative loss bounds for on-line density estimation with the exponential family of distributions," *J. of Machine Learning Research*, vol. 43, no. 3, pp. 211–246, Jun. 2001.

[18] V. Vovk, "Competitive on-line statistics," *International Statistical Review*, vol. 69, no. 2, pp. 213–248, 2001.

[19] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *J. of Machine Learning Research*, vol. 5, pp. 101–141, 2004.

[20] Y. Chen and X. Ye, "Projection onto a simplex," University of Florida, Tech. Rep., Jan. 2011.

[21] N. Cesa-Bianchi *et al.*, "Incremental algorithms for hierarchical classification," *J. of Machine Learning Research*, vol. 7, pp. 31–54, 2006.

[22] N. Cesa-Bianchi, C. Gentile, *et al.*, "Robust bounds for classification via selective sampling," in *Proc. of the ICML*, 2009, pp. 121–128.

[23] J. Nagi, F. Ducatelle, G. A. Di Caro, *et al.*, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *Proc. of the IEEE ICSIPA*, 2011, pp. 342–347.