Online Multiple Instance Learning Applied to Hand Detection in a Humanoid Robot

Ciliberto, Carlo; Smeraldi, Fabrizio; Natale, Lorenzo; Metta, Giorgio

ISSN 2043-0167

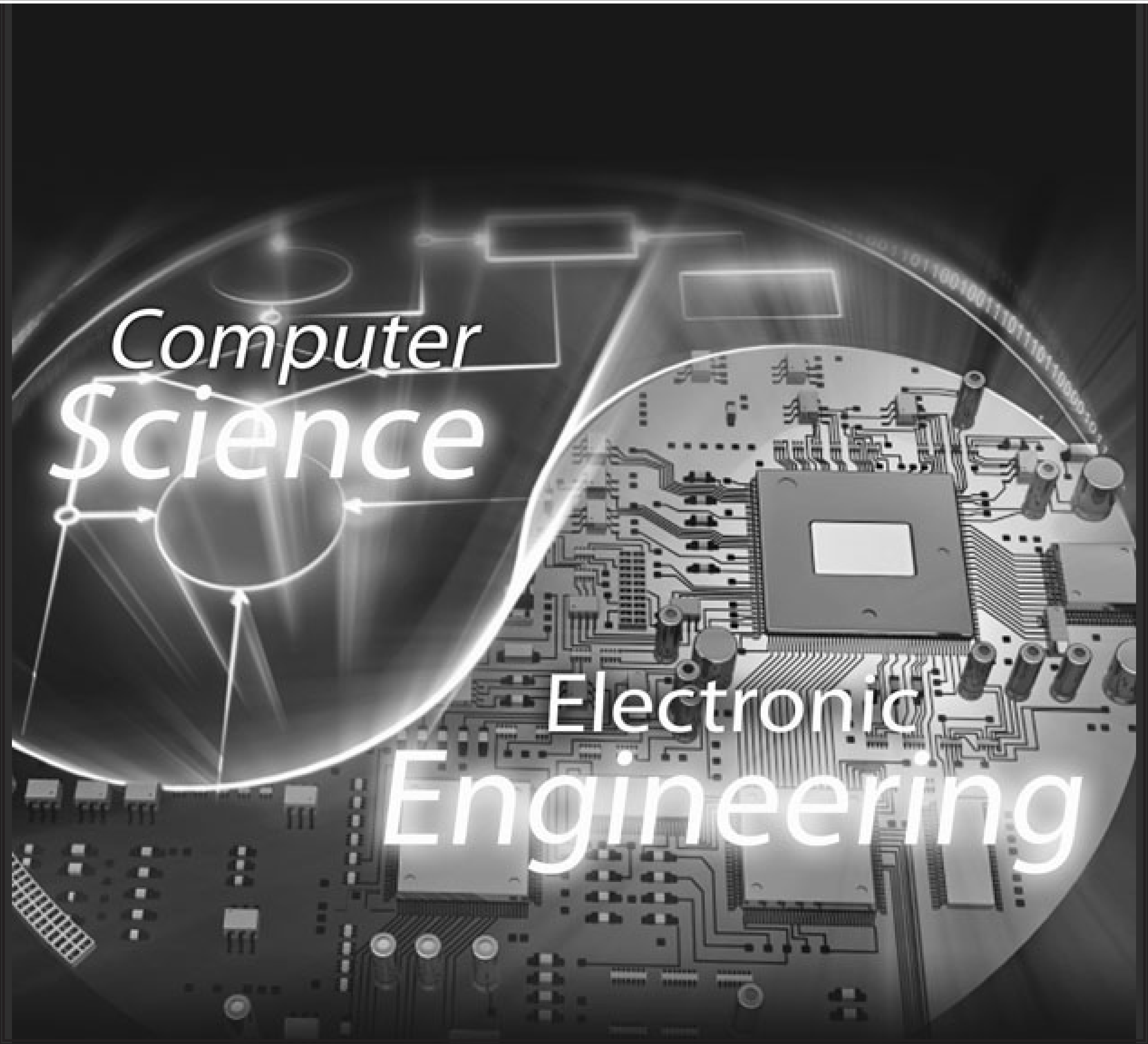
Online Multiple Instance Learning Applied to Hand Detection in a Humanoid Robot

Carlo Ciliberto, Fabrizio Smeraldi, Lorenzo Natale and Giorgio Metta

Queen Mary
University of London
School of Electronic Engineering
and Computer Science

EECSRR-10-03

June 2010



Online Multiple Instance Learning Applied to Hand Detection in a Humanoid Robot

Carlo Ciliberto
Robotics, Brain and Cognitive Sciences
Italian Institute of Technology
Via Morego 30, 16163 Genova GE, Italy
carlo.ciliberto@iit.it

Fabrizio Smeraldi
School of Electronic Engineering
and Computer Science
Queen Mary University of London
Mile End Road, London E14NS, UK
fabri@dcs.qmul.ac.uk

Lorenzo Natale
Robotics, Brain and Cognitive Sciences
Italian Institute of Technology
Via Morego 30, 16163 Genova GE, Italy
lorenzo.natale@iit.it

Giorgio Metta
Robotics, Brain and Cognitive Sciences
Italian Institute of Technology
Via Morego 30, 16163 Genova GE, Italy
lorenzo.natale@iit.it

Abstract

Algorithms with limited supervision requirements are particularly useful for intelligent autonomous agents, that must be able to learn with minimal supervision and adapt online to a changing environment. We propose an online Multiple Instance Learning algorithm based on boosting and apply it to the problem of visual detection of the hand of a humanoid robot. Our approach solves the multiple instance problem at the level of the weak learners, allowing the detection of objects represented by more than one positive instance. Feature selection and adaptation are performed online as new data are fed to the algorithm. Experiments in real-world conditions on an iCub humanoid robot confirm that the algorithm can learn the appearance of the hand, reaching an accuracy comparable with off-line algorithms. This remains true when supervision is generated by the robot itself in a completely autonomous fashion. To the best of our knowledge, this is the first implementation of online multiple-instance learning on a robotic platform.

1 Introduction

On-line adaptation is an essential capability for artificial cognitive agents operating in the real world. As the scientific community devotes growing attention to the development of such systems, there is a growing demand for learning techniques in which data is acquired and used on-line and largely autonomously.

Ideally, training data should be collected automatically without human supervision. Learning in real world applications is therefore hampered by an unfavorable tradeoff between the accuracy of the training examples and their availability. One way to relax the requirements on supervision is to adopt a learning paradigm such as Multiple Instance Learning (MIL). In this framework, training examples come in "bags" that contain positive and negative instances sharing a common label; a positive bag contains at least one positive instance, while a negative bag is guaranteed to contain no positive instances at all.

In this work, we develop an online MIL algorithm that uses an online variant of Adaboost [1, 2] to combine a family of weak hypotheses specifically designed for MIL [3]. As the online algorithm does not make provisions for feature selection, we implement this by wrapping weak learners inside

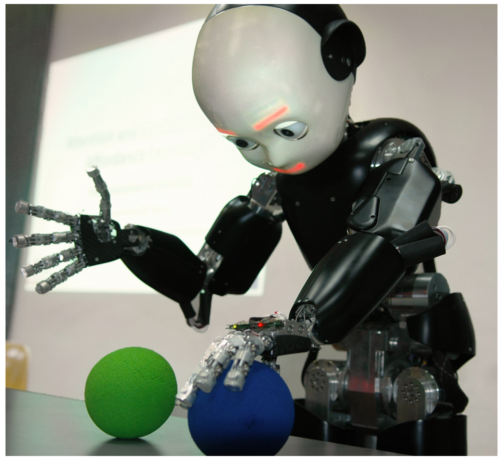


Figure 1: The hardware platform: the iCub robot

selectors, similarly to what is done in [4]. We apply our algorithm to a visual learning problem, namely hand detection in the iCub humanoid robot (Figure 1). Our algorithm only requires knowledge of whether the hand is present in the visual field for training. This enables the robot to learn the appearance of the hand without any supervision other than a self-generated coarse labelling based on the co-occurrence of motion in the visual stream and in the motors. Experiments are performed in a challenging scenario using images coming from the embarked cameras while the robot is operating in realistic, cluttered environment. The resulting classifier performs reliably, with error rates comparable to the batch version of the algorithms.

We are aware of only another online MIL algorithm in the literature [5], also based on a variant of boosting. Our approach differs in that the Multiple Instance nature of the problem is dealt with at an early stage by the weak learners rather than at the level of the ensemble method. This allows the classification of targets described by more than one positive instance (see Section 3). Implementation on a robotic platform has not, to the best of our knowledge, been reported before.

2 Previous work

Hand detection: Visual localisation of the end-point is crucial for closed loop control of robotic manipulators [6]. Normally this problem is solved by employing markers that greatly simplify the detection. This approach, however, has clear limitations and can be applied only in controlled settings. In humanoid robotics, there have been recently some attempts to solve the problem of hand detection in a generic way [7–9] . These works share a common idea in that they integrate vision with the arm joints state to perform the autonomous visual segmentation of the hand. In these cases, however, the main concern is to investigate to what extent the integration between vision and motorial information can help solve the hand detection task. These works generally employ simple visual descriptors (for example [8] uses a colour histogram); arguably better result could be expected by using more sophisticated and robust visual features (e.g. [10, 11]).

Multiple Instance Learning: The MIL approach originated in a bio-informatics setting in the late Nineties [12, 13], when the Diverse Density algorithm was developed [14], and was quickly applied to object recognition. In our terms, MIL corresponds to a scenario in which training images (that play the role of the bags of instances) are identified as either containing or not containing the object of interest, without its location and size being specified. Interest in this technique has been renewed more recently with the development of SVM-based algorithms such as DD-SVM [15] and MILES [16].

Of special interest to us are the boosting-based approaches [3, 17, 18], that lend themselves more naturally to modifications for on-line learning because of the iterative nature of boosting algorithms. These approaches differ in the particular flavour of boosting used as well as in the way that the MIL paradigm is implemented. A variant of the Linear Programming Boosting framework (LPBoost) is used in [17]. At each iteration of LPBoost, a linear programming problem is solved to maximise the margin of the training examples. The MIL generalisation relaxes this requirement by optimising to achieve a large margin for at least one of the patterns in each bag. In [18] the AnyBoost framework is used. Weak learners classify single instances, and the probability that a bag is positive is obtained by the Noisy-OR of the probability of each instance being positive. The target function optimised by AnyBoost is the likelihood of each bag being positive. The only implementation of online MIL of which we are aware builds on this approach, by introducing a variant of AnyBoost [5]. Weak learners based on Haar-like features are trained in an online fashion. For each new training example however, a new strong classifier is estimated from scratch by the boosting algorithm.

The above boosting approaches, with the exception of [3], share a common "late" handling of the MIL problem: the weak learners act on single instances, and the multiple instance problem is handled by the ensemble algorithm. By contrast, in [3] an "early" approach is adopted: the weak learners themselves, defined as balls of optimal radius in feature space, directly classify bags of instances as opposed to single instances. By dealing with MIL complexities at the level of the weak learners, this algorithm allows using a standard implementation of two-class Adaboost (or indeed, of any equivalent variant). As detailed below, we choose this approach because of its flexibility and of the compatibility of the particular type of weak learners with our choice of visual descriptors.

3 Contributions

We propose an online implementation of Multiple Instance Learning for the purpose of semi-supervised recognition of a robot hand. The MIL nature of the problem is derived from application: the robot does not initially know the appearance or the location of the manipulator in the visual field, but it can control the motors to bring the manipulator in view or out of view.

We extract from the image a set of interest points on which we compute SURF descriptors [11]. SURF are robust to scale and orientation variations and have been empirically proved [19] to be remarkably faster than SIFT [10], which makes them more suitable for online processing. Descriptors are extracted from the whole image; thus the object of interest (the hand) is represented by a group of relevant descriptors (instances) embedded in a larger positive bag, that also includes interest points from the background. This differs from [5, 18] where the object is represented by a single instance.

For the above reasons, it is crucial that the weak learner themselves are able to deal with multiple instances; we therefore follow the boosting-based approach of Auer et al. [3], as their framework involves weak learners that directly classify bags rather than single instances. Also, the specific nature of the weak learners they propose (detailed below in Section 4.2) makes them more suitable for classifying high-dimensional descriptors such as SURF.

In order to adapt the algorithm for online use, we replace the standard AdaBoost algorithm used in [3] with the online boosting procedure proposed in [1], that has already been used with success in vision applications [4]. While Oza's algorithm can be trained more efficiently than the variant used in [5], it has no provisions for feature selection. This limitation has been dealt with in [4] by introducing *selectors*, that are essentially wrappers for the weak learners that allow selecting the hypotheses that perform best. We follow this scheme, apply it to the MIL problem and extend it to the case that not all the weak learners are known from the start (see Section 4.3).

Finally, we present an implementation of our algorithm on an iCub humanoid robot and use it to produce the experimental results reported in Section 5. The code has been submitted to the iCub repository [20] and is available to all researchers using the open-source robotic platform.

4 Multiple Instance Learning and Online Boosting

4.1 AdaBoost and online boosting

AdaBoost is a well established offline boosting algorithm that adopts a greedy strategy to combine a series of inaccurate weak classifiers into a highly precise *strong classifier* [2]. It does so by maintaining a distribution of weights $\mathbf{\Lambda}$ over the training set. At each iteration, the weak learner with the lowest misclassification rate with respect to $\mathbf{\Lambda}$ is added to the strong classifier with a coefficient

Online Boosting

Initialization.

- Let $\mathcal{H} = \{h_1 ,..., h_N\}$ be a set of weak classifiers with Learning Principle L.
- Set $\lambda_n^w = \lambda_n^c = 0 \ \ \forall n \in \{1,...,N\}$.

Training.

At each iteration step t a novel sample I_t is presented to the system:

- Set the importance weight of the sample to $\lambda = 1$.
- For $n \in \{1,...,N\}$ do:
  1. update $h_n \leftarrow L(h_n, I_t, \lambda)$
  2. if h_n correctly classifies I_t:
     - $\lambda_n^c \leftarrow \lambda_n^c + \lambda$; $\ \epsilon_n = \frac{\lambda_n^w}{\lambda_n^c+\lambda_n^w}$; $\ \lambda \leftarrow \lambda \frac{1}{2(1-\epsilon_n)}$

     else:
     - $\lambda_n^w \leftarrow \lambda_n^w + \lambda$; $\ \epsilon_n = \frac{\lambda_n^w}{\lambda_n^c+\lambda_n^w}$; $\ \lambda \leftarrow \lambda \frac{1}{2\epsilon_n}$
  3. Define the relevance weight of the n-th weak learner as $\alpha_n = \log\left(\frac{1-\epsilon_n}{\epsilon_n}\right)$
- end.

Strong Classifier.

After every learning iteration, the score assigned by the strong classifier to a bag $I \in \mathcal{I}$ is:

$$S(I) = \sum_{n=1}^{N} \alpha_n \cdot h_n(I) \ .$$

Table 1: Oza's online AdaBoost

dependent on its accuracy. The weights are subsequently updated so that misclassified training examples become more important at the next iteration.

The main obstacle to an online formulation of the algorithm is the need to keep track of a weight distribution over a training set that is constantly growing. In the online variant introduced by Oza [1] the iterative structure of the algorithm is retained, but the examples are now propagated down a pre-ordered, fixed list of weak learners that make up the strong classifier (Table 1). Each weak learner increases (decreases) the weight of the samples it misclassifies (classifies correctly) before passing them on to the next weak learner.

Finally, each weak learner keeps track of its error rate based on the weight of the samples it classifies. Because the weak learners are fed the training examples one at a time, an online *Learning Principle* L needs to be specified for them.

4.2 MIL and Boosting

Auer and Ortner [3] proposed to combine MIL and boosting from the perspective of high dimensionality features. In their framework, a weak learner is a ball B in the feature space $\mathbb{R}^N$. If we denote a bag of instances by $I = \{x_i\}$, a ball B classifies as positive the bags I such that $I \cap B \neq \emptyset$. Under these assumptions, given a training set $\mathcal{I}$ over which a weight distribution $\mathbf{\Lambda}$ has been provided, the quality of any classifier B can be assessed by evaluating its *distribution accuracy* $D(B, \mathbf{\Lambda})$, i.e. the sum of all the weights $\mathbf{\Lambda}$ associated to the training bags correctly classified by B.

In the original work, classical Adaboost is applied to the set of weak classifiers represented by the balls $\{B_r(x)\}$ centred on every positive instance x in the training set. Weak learners are trained by optimising their radius according to $r = \operatorname{argmax}_{r'>0} \ D(B_{r'}(x), \mathbf{\Lambda})$; however, this is not directly feasible in an online context as training data are provided to the algorithm in a sequential fashion.

Online MIL weak learner.

Definition. An *online MIL* weak learner is a pair $h = (B_r(x), \mathbf{\Lambda})$ associated to a weighted distribution $\mathbf{\Lambda}$ and to a ball $B_r(x) \subset X$ centered on a positive instance x .

Classification.

$$\forall I \in \mathcal{I} \qquad h(I) = \begin{cases} 1 & \text{if } I \cap B_r(x) \neq \emptyset \\ -1 & otherwise \end{cases}$$

Learning Principle. For any pair $(I, \lambda) \in \mathcal{I} \times \mathbb{R}_+$ and weak classifier $h = (B_r(x), \lambda)$, the Learning Principle $L(h, I, \lambda)$ is defined as follows (see Figure 3 Left):

- update $\mathbf{\Lambda} \leftarrow \mathbf{\Lambda} \cup \{(I, \lambda)\}$.
- if $|\mathbf{\Lambda}| > n_{max}$, eliminate from $\mathbf{\Lambda}$ the pair (I, λ_{min}) with minimum weight λ_{min}.
- compute $\tilde{r} = argmax_{r>0} \;\; D(B_r(x), \mathbf{\Lambda})$.
- return the updated weak classifier $h = (B_{\tilde{r}}(x), \mathbf{\Lambda})$.

Table 2: Definition of online MIL weak learner

In Table 2 we propose an adaptation of Auer and Ortner's ball learners to the online framework presented in Section 4.1. The main difference is in the Learning Principle: whenever a novel training bag arrives, the radius is updated to keep the distribution accuracy maximized. However, as new data comes in, training samples with the lowest weights assume less and less importance and can be discarded to avoid memory overstress.

4.3 Weak learner selectors

MIL over a continuous data stream can in principle be achieved by applying the online boosting algorithm described in Section 4.1 to the weak learners introduced in Section 4.2 above.

However, in an online context it is likely that useful and descriptive features (and hence potential centres for new weak classifiers) will not be available from the start, but may become available, for instance, as the object to be learned rotates and some of its previously hidden parts become visible. However, the algorithm in Table 1 has no way to access such information, as it requires the set of weak learners to be defined a priori.

A possible solution is to employ a class of more general weak learners: the *selectors*. These were originally introduced in [4] to approach feature selection problems via online boosting. A selector acts as a wrapper for a pool of weak learners (Table 3). Whenever a training sample arrives, the selector updates all the weak learners in its pool and marks the one with lowest error rate as its current decision function. At the same time, the element that exhibits the worst classification performance is substituted with a new one, selected at random from a weak learner cache that stores all weak classifiers corresponding to the instances in the positive bags seen so far. This allows the boosting algorithm to access novel weak learners as they are needed.

5 Hand Detection

5.1 Robotic Platform

The experiments described in this paper were carried out on an iCub humanoid robot (see Figure 1). The iCub is a complete humanoid robot, with 53 degrees of freedom [21]. This platform was designed to study manipulation and for this reason particular attention has been devoted to the design of the hands (overall each arm has 16 degrees of freedom). Six motors actuate the head: three control the head at the level of the neck, whereas other three control two cameras around a common tilt axis and two independent pan axes. The sensory system includes two cameras for vision, an inertial sensor, force and position feedback from all the motors (optical encoders).

Selector

Definition. A selector is a couple $s = (P, \bar{m})$ where $P = \{(h_1, \epsilon_1)\ ,...,\ (h_M, \epsilon_M)\}$ is a set (or pool) of weak learners h_i with associated error rate ϵ_i and $\bar{m}$ is the index of the weak learner currently chosen by the selector

Classification.

$$\forall I \in \mathcal{I} \qquad s(I) = h_{\bar{m}}(I)\ .$$

Learning Principle. For any couple $(I, \lambda) \in \mathcal{I} \times \mathbb{R}_+$ and selector $s = (P, \bar{m})$, the learning rule $L(s, I, \lambda)$ is defined as follows (see Figure 3 Right):

- For $i \in \{1\ ,...,\ M\}$ do:
  - update $h_i \leftarrow L^h(h_i, I, \lambda)$ where L^h is h's Learning Principle.
  - update the error rate ϵ_i (as in step 2. of Table 1) according to $h_i(I)$.
- set $i_{min} = \text{argmin}_i\ \epsilon_i$ and $i_{max} = \text{argmax}_i\ \epsilon_i$.
- substitute $h_{i_{max}}$ with a new weak learner chosen at random and set $\epsilon_{i_{max}} = 0$.
- return the updated selector $s = (P, i_{min})$

Table 3: Definition of a weak learner selector

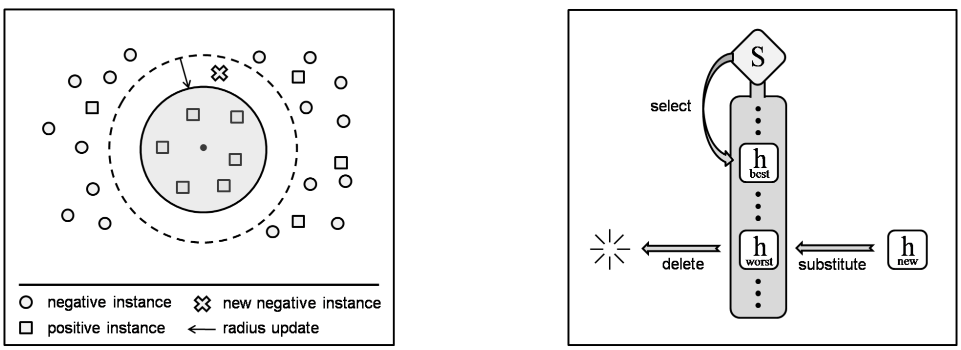
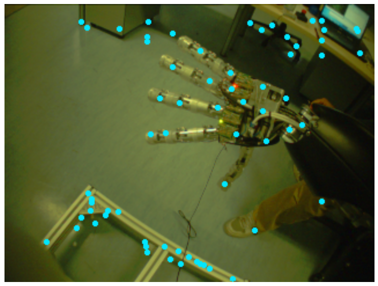


Figure 2: **Learning Principles**. (Left) Example of a learning step for an online MIL weak learner: a new negative instance has occurred (the cross) and the radius is consequently reduced to keep the distribution accuracy optimized. (Right) a typical learning iteration for a selector: the best weak learner is chosen as current decision function while the worst one is substituted with a completely new one randomly extracted from the weak learner cache.

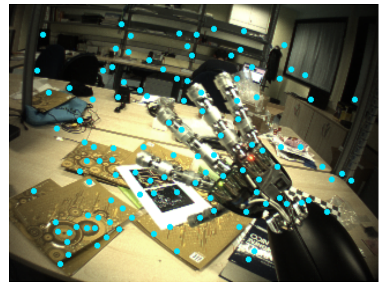
The iCub is an open system, the design and documentation of hardware and software are licensed under the Free Software Foundation licenses. All parts of the system can be freely replicated and customized; at the moment of writing several copies of the robot exist in research laboratories around the world (mostly in Europe). Researchers working on these robots have access to a large repository of software [20] which contains the results of the work of other laboratories, including the work presented in this paper.

5.2 Experiments

We validate the online MIL boosting framework by testing it on the hand detection problem. The system is trained using the image stream from the right eye camera, acquired while the robot performed random gaze shifts and right arm movements. Gaze shifts involved motion of the head and torso and increased background and of illumination variability. The frequency of hand occurrences in the data was controlled and kept around 50%. Furthermore the wrist was controlled so that the back of the hand faced the cameras and had approximately constant orientation.



(a) Uniform background.



(b) Cluttered background.

Figure 3: SURF features extracted from images depicting two different kind of backgrounds.

We recorded two datasets, each consisting in a sequence of 1500 frames separated in a training set (the first 500 frames) and a test set (the remaining 1000 frames). The datasets were recorded in different settings by changing the type of background and according to different labeling strategies:

Sequence 1 (**uniform**): In this case the robot operated in a relatively simple environment. The hand generally appears on a uniform, uncluttered background (Figure 3(a)). Labels are assigned manually by a human observer.

Sequence 2 (**cluttered**): Feature-rich distractors are explicitly added to the environment, so that the features extracted from each image have a much larger probability of belonging to the background (Figure 3(b) shows a typical image). Two series of labels are available. Images are labelled autonomously by the robot during acquisition by detecting co-occurrence of movement in the visual field and motion in the motor encoders. This labeling strategy is intrinsically imprecise, leading to a 10% of erroneous training labels. A second series of labels is assigned by a human observer.

We measure the performance of our online MIL classifier on the two datasets by determining the Equal Error Rate (EER). Results are averaged across 20 runs in order to account for the random substitution of new weak learners. We also test the dependence of the algorithm to cope with the order of appearance of the samples; for this purpose we trained the classifier in two ways: 1) on the natural sequence of the samples (i.e. as they were acquired by the robot) and 2) by randomly shuffling the training samples (averaged across 20 runs). Finally, we compare the performance of the online classifier with the the batch MIL algorithm of Auer and Ortner [3].

Experimental results are summarized in Table 4, that lists for each experiment the average EER across 20 runs and the standard deviation. In all the experiments the number of selector was set to 35, each of 50 elements. These numbers were chosen empirically.

Figure 4 also shows a more detailed comparison between the automatic and the manual labeling strategies performed on the **cluttered** dataset. The figure shows the EER as a function of the number of training examples when the classifier is trained online using automatic and manually labelled data respectively. In this case the order of the samples was untouched. As before training was performed 20 times, and average and standard deviation of the EER computed for each number of training samples. In both cases performance increases as more samples are available.

Our algorithm consistently achieves a good performance even in presence of a cluttered environment. The EER is comparable with the ones of the batch version. Of particular importance for our application is that the classifier maintains good performance when labels are assigned autonomously by the robot (which introduces mistakes in the training set), and irrespectively of the order of presentation of the samples (this is especially important because it shows that the algorithm can be used online and it does not require randomization of the training set).

6 Conclusions

We presented an on-line MIL algorithm based on a variant of Adaboost. Our algorithm tackles the MIL problem at the level of the weak learners and includes a mechanism for online feature selection. We validated our algorithm with an application to the problem of hand detection on a humanoid

Dataset	Labelling	Order	EER Online	EER Offline
uniform	Manual	Natural	(9.0 ±0.7)%	7.3%
uniform	Manual	Shuffled	(7.4±2.5)%	7.3%
cluttered	Manual	Natural	(13.0±1.9)%	7.5%
cluttered	Manual	Shuffled	(12±1)%	7.5%
cluttered	Auto	Natural	(18±2)%	9.9%
cluttered	Auto	Shuffled	(19±5)%	9.9%

Table 4: Results of experiments conducted over the two datasets. The table reports the average EER of our algorithm (Online) and the offline MIL adaboost (Offline). Labeling strategies are identified with *manual* (human supervision) and *auto* (automatic labeling by the robot). We varied the order of presentation of the samples. With *natural* we identify the cases in which the order of the samples was untouched, with *shuffled* those in which the order was randomly modified.

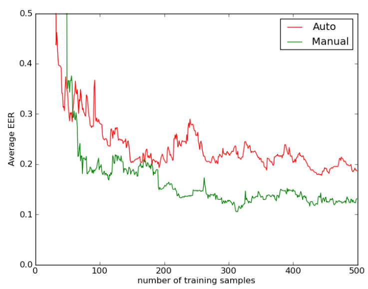


Figure 4: EER vs number of training samples: cluttered background, automatic vs manual labelling, online MIL.

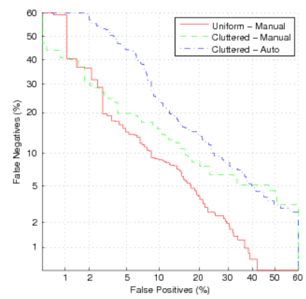


Figure 5: ROC curves for three background and labeling choices , online MIL algorithm.

robot, using SURF descriptors to encode the salient points of the visual scene. We implemented the algorithm on an iCub platform, that we used to run experiments in realistic conditions. These showed that online MIL boost performs consistently well and comparably with the equivalent offline algorithms, even in a cluttered visual environment.

The reduced demands of MIL algorithms in terms of the type of supervision provided to the system allowed the robot to generate the supervision signal autonomously by exploiting a combination of visual motion and motor activity to label images coming from the cameras; even in this case the system behaved reliably, in spite of the erroneous labels contained in the training set.

To the best of our knowledge, this is the first application of online MIL in robotics. In this paper we tested our algorithm on the specific problem of hand detection; however, the MIL nature of the approach opens up a wide range of applications in cognitive robotics. In our experiments labeling was autonomously generated by detecting co-occurrence of events in different sensory systems (vision and motor system); the same algorithm could in principle be applied in contexts where supervision is provided by integrating sensory modalities like vision, touch and sound (or speech). Also, the flexibility of the boosting algorithm makes it easy to integrate different features in the classification process.

References

[1] N. C. Oza and S. Russell. Online bagging and boosting. In Proceedings of the 8th International Workshop on Artificial Intelligence and Statistics, pages 105–112, January 2001.

[2] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Science, 55(1), 1997.

[3] P. Auer and R. Ortner. A boosting approach to multiple-instance learning. In Proceedings of the 15th European Conference on Machine Learning, Pisa (Italy), pages 63–74, 2004.

[4] H. Grabner and H. Bischof. On-line boosting and vision. In Proceedings of CVPR, volume 1, pages 260–267, 2006.

[5] B. Babenko, Ming-Hsuan Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In CVPR, 2009.

[6] Seth Hutchinson, Greg Hager, and Peter Corke. A tutorial on visual servo control. IEEE Transactions on Robotics and Automation, 12:651–670, 1996.

[7] L. Natale, F. Orabona, G. Metta, and G. Sandini. Sensorimotor coordination in a baby robot: learning about objects through grasping. In C. Von Hofsten and K. Rosander, editors, Progress in Brain Research, From Action to Cognition, volume 164, pages 403–424. Elsevier, 2007.

[8] C. Kemp and A. Edsinger. What can I control?: The development of visual categories for a robot's body and the world that it influences. In International Conference on Epigenetic Robotics, Paris, France, September 2006.

[9] K. Gold and B. Scassellati. Using probabilistic reasoning over time to self-recognize. Robotics and Autonomous Systems, 57(4):384–392, 2009.

[10] David G. Lowe. Object recognition from local scale-invariant features. In ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2, page 1150, Washington, DC, USA, 1999. IEEE Computer Society.

[11] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In 9th European Conference on Computer Vision, Graz Austria, May 2006.

[12] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple-instance problem with axis-parallel rectangles. Artificial Intelligence, 89(1-2):31–71, 1997.

[13] P. Auer. On learning from multiple-instance examples: empirical evaluation of a theoretical approach. In NeuroCOLT Technical Report Series, NC-TR-97-025, March 1997.

[14] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In Advances in Neural Information Processing Systems, volume 10, pages 570–576, 1998.

[15] Y. Chen and J. Z. Wang. Image categorization by learning and reasoning with regions. Journal of Machine Learning Research, 5:913–939, 2004.

[16] Y. Chen, J. Bi, and J. Z. Wang. MILES: Multiple instance learning via embedded instance selection. IEEE Transactions on PAMI, 28(12):1931–1947, 2006.

[17] S. Andrews and T. Hofmann. Multiple instance learning via disjunctive programming boosting. In Advances in Neural Information Processing Systems, volume 16, 2004.

[18] P. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. In Advances in Neural Information Processing Systems, volume 18, 2005.

[19] J. Bauer, N. Sunderhauf, and P. Protzel. Comparing several implementations of two recently published feature detectors. In International Conference on Intelligent and Autonomous Systems (IAV), Toulouse (France), 2007.

[20] The icub web site. `http://www.icub.org`.

[21] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori. The iCub humanoid robot: an open platform for research in embodied cognition. In Workshop on Performance Metrics for Intelligent Systems, National Institute of Standards and Technology, Washington DC, USA, August 19-21 2008.