Upper Confidence Weighted Learning for Efficient Exploration in Multiclass Prediction with Binary Feedback*

Hung Ngo, Matthew Luciw
IDSIA, Galleria 2
Manno-Lugano 6928
Switzerland
{hung,matthew}@idsia.ch

Ngo Anh Vien
MLR Lab
University of Stuttgart
70569 Stuttgart, Germany
ngovn@ipvs.uni-stuttgart.de

Jürgen Schmidhuber
IDSIA, Galleria 2
Manno-Lugano 6928
Switzerland
juergen@idsia.ch

Abstract

We introduce a novel algorithm called Upper Confidence Weighted Learning (UCWL) for online multiclass learning from binary feedback. UCWL combines the Upper Confidence Bound (UCB) framework with the Soft Confidence Weighted (SCW) online learning scheme. UCWL achieves state of the art performance (especially on noisy and non-separable data) with low computational costs. Estimated confidence intervals are used for informed exploration, which enables faster learning than the uninformed exploration case or the case where exploration is not used. The targeted application setting is human-robot interaction (HRI), in which a robot is learning to classify its observations while a human teaches it by providing only binary feedback (e.g., right/wrong). Results in an HRI experiment, and with two benchmark datasets, show UCWL outperforms other algorithms in the online binary feedback setting, and *surprisingly* even sometimes beats state-of-the-art algorithms that get full feedback, while UCWL gets only binary feedback on the same data.

1 Introduction

Consider an interactive classification learning scenario between a human and a robot as roughly sketched in Figure 1. Every interaction proceeds as follows. The robot receives an observation, e.g., the human shows it an object, a gesture, etc. Next, the robot outputs a class label, from a set of known labels (all possible labels are known beforehand). Then, the human might provide feedback to the robot regarding the accuracy of the label. This feedback is in form of a binary signal (either True or False). The human's feedback will usually, but not always, be accurate. Upon receiving the feedback, the robot updates its learning model. The robot should improve its classification accuracy as much as possible, with as little feedback as possible. As this is a real-time setting, keeping the response time of the robot low is also important.

*This work was funded through the 7th framework program of the EU under grants #231722 (IM-Clever project), #288551 (WAY project) and #270247 (NeuralDynamics project).

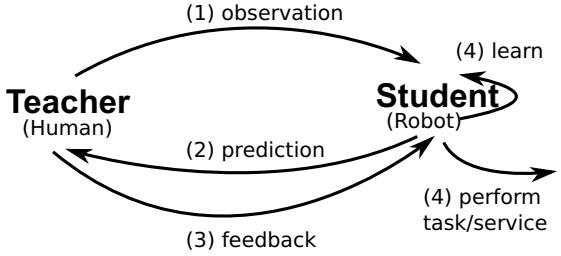

Figure 1: The general human-robot interactive learning scenario.

This general setting allows interactions to be more realistic in some ways compared to straightforward applications of typical learners. Here, there are no distinct training and testing phases, nor is the human required to provide feedback for each observation. Additionally, binary feedback is more flexible than full feedback, since, in the most general case, the feedback itself must be communicated to the robot, which amounts to another classification problem. Pre-defining two classes that can be reliably sensed (such as nodding the head for yes or shaking the head for no) is quite feasible, but communicating a large number of different class labels requires a direct connection to the learning algorithm, and requires more attention from and is a larger burden upon the human.

To implement the above setting, one needs to deal with several issues that will stymie typical learning algorithms. First, the observation sequence comes in an online manner, and few or even no statistical assumptions can be made about the underlying process generating the data — the human might even select the observation sequence in an adversarial manner. This issue implies that traditional statistical learning methods, which rely on assumptions about the data generating process (e.g., i.i.d), are not directly applicable. The competitive online learning framework [Cesa-Bianchi and Lugosi, 2006; Vovk, 2001; Azoury and Warmuth, 2001], which aims to make as few mistakes as possible on *any* sequence of examples, is more appropriate.

Second, there is an implicit exploration-exploitation tradeoff — the robot may not necessarily want to select its current *best* prediction, in favor of the *potentially most informative* prediction — e.g., that is expected to lead to useful feedback, in order to accelerate learning progress. The exploration-

exploitation issue casts this problem as a so-called "contextual bandit" [Kakade *et al.*, 2008], and links the problem to reinforcement learning [Szepesvári, 2010]. A promising approach for dealing with exploration-exploitation is the Upper Confidence Bound (UCB; [Auer, 2003]) framework, which combines predicted correctness scores and uncertainty estimates, and use this to select the output. The robot can *explore* by selecting uncertain answers, and will (probably) receive feedback so that the uncertainty decreases. Answers that have intrinsically high uncertainty (i.e., due to noise in the process) will eventually have low scores, and should not be selected after some time. In this sense, one can even consider this a form of artificial curiosity [Schmidhuber, 2010].

Third, and most importantly, a learning algorithm that must deal with our imagined scenario must effectively learn from *binary* feedback, instead of the standard *full* feedback (the true class label). Binary feedback is obviously less informative than full feedback, and one would expect this leads to a higher sample complexity compared to full feedback learners. We note that it is not trivial to extend a good full feedback algorithm to a good binary feedback algorithm — it is known that one must include exploration in the online learning and incomplete feedback setting [Kakade *et al.*, 2008; Crammer and Gentile, 2011], and recent work has shown that the UCB framework as state-of-the-art in this regard. But to implement the UCB framework, the confidence information must be estimated in an online manner, from the binary feedback. It's not trivial to derive that information for an online algorithm, without any assumptions about the distribution.

The algorithm we introduce here, Upper Confidence Weighted Learning (UCWL), uses the UCB framework for the online learning with binary feedback setting. In this sense, it is like the recent Confidit algorithm [Crammer and Gentile, 2011]. But unlike Confidit, which maintains a linear classifier and calculates confidence (inverse uncertainty) information through usage of a regularized least squares (RLS) linear method, UCWL applies a state-of-the-art method from online learning for linear classification, inspired by the recently proposed Confidence-Weighted (CW) online learning algorithms [Crammer *et al.*, 2008; Wang *et al.*, 2012; Crammer *et al.*, 2009b; Orabona and Crammer, 2010]. We focus on a particular family of linear classifiers that maintains a multivariate Gaussian distribution over the weight vectors. From each training example, the learning model is updated aggressively, while maintaining the knowledge learned so far by not changing too much, in the Kullback-Leibler (KL) divergence sense, from the previous model. UCWL is formulated for the multiclass setting. UCWL has closed-form, adaptive large-margin style update rules. We derive *per-instance* confidence intervals for each prediction margin, from which optimistic predictions are made and the exploration-exploitation tradeoff is handled in an informed way.

Results on three datasets, one designed to emulate the Human Robot Interaction (HRI[1]) scenario (our eventual target application), and two other benchmarks for evaluating and comparing the performance, show that UCWL achieves excellent performance, outperforming state-of-the-art algorithms in the same binary feedback setting. Additionally, although using only binary feedback, UCWL approaches the online mistake rate of the best algorithm and even sometimes (in the presence of label noise) outperforming some other algorithms running in the *full* feedback setting, on the same sequence of observations.

[1]To the best of our knowledge, ours is the first method applied to HRI for multiclass online learning with binary feedback.

2 Background and Related Work

Online Learning with Binary Feedback. The online classifier learning with binary feedback setting was formulated in terms of an exploration-exploitation tradeoff by researchers from the reinforcement learning community, who defined the class of *contextual bandit* problems. The Banditron [Kakade *et al.*, 2008] is a primal example. But Banditron's exploration is randomized (uninformed), so its performance is suboptimal. Confidit [Crammer and Gentile, 2011], a multi-class classification scheme improved upon Banditron via the maintenance and use of upper confidence bounds for informed exploration. With UCB, the label selection mechanism takes the maximum combination of *score* (predicted label) and *uncertainty*. In Confidit, the ℓ^2-regularized least squares algorithm is used — by maintaining a data correlation matrix, the confidence information associated with each prediction can be derived and used as the needed uncertainty.

Online Learning for Linear Classification. Online learning algorithms [Cesa-Bianchi and Lugosi, 2006; Vovk, 2001; Azoury and Warmuth, 2001] are a family of actively studied machine learning techniques that can be applied to the interactive learning scenario. An intuitive working principle in online learning algorithms is to balance the two conflicting goals in making an update of their learning model: the updated model must give better prediction (i.e., smaller loss) on the current example while not forgetting much of the information it has acquired in the preceding interactions (i.e., small divergence with the old model) [Azoury and Warmuth, 2001; Crammer *et al.*, 2006]. For example, the state-of-the-art first order online learning algorithm Passive-Aggressive (PA) [Crammer *et al.*, 2006] finds a new weight vector closest in ℓ^2-norm sense to the old one, under the constraint that its hinge loss on the current example is zero.

Recently, second-order online learning algorithms using linear models have shown to achieve state-of-the-art performance on many online learning tasks. These algorithms maintain extra confidence information, either on the learning weight vectors [Crammer *et al.*, 2008], or on the prediction margins [Cesa-Bianchi *et al.*, 2005], then exploit this information to guide and adapt the online learning process. The CW learning algorithms [Crammer *et al.*, 2008] and its follow-up algorithms AROW [Crammer *et al.*, 2009b] and SCW [Wang *et al.*, 2012] maintain a multivariate Gaussian distribution over the learning weights of a linear classifier. These algorithms use the input data to update the distribution parameters, and utilizes this confidence information to decide the direction and magnitude of the weight updates. The soft-margin extension of CW, SCW, is the first second-order online learning algorithm possessing all four salient properties

of i) large margin training, ii) confidence weighting, iii) capability to handle noisy and non-separable data, and iv) adaptive margin requirement. UCWL is developed based on the SCW algorithm, which it extends to multiclass prediction and provides a systematic, efficient exploration mechanism for dealing with the binary feedback scenario.

3 Confidence-Weighted Adaptive Large-Margin Learning

The algorithms summarized in this section are the basis for developing UCWL. They only consider binary classification with full feedback. In next section we derive UCWL as an extension to the multiclass with binary feedback setting.

3.1 Confidence-Weighted Learning (CW)

CW [Crammer *et al.*, 2008] is motivated from the insight that given the set of features in an instance, low-confidence feature weights should be updated more aggressively than high-confidence ones. This idea is realized by modeling confidence in learning weights of a linear classifier using a multivariate Gaussian distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^d$, and covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$.

Conceptually, on receiving an instance $\mathbf{x}_t \in \mathbb{R}^d$ at round t, CW classifier draws a weight vector $\mathbf{w}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$ and predicts the corresponding label as $\mathsf{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$; the absolute value of the prediction margin $|\mathbf{w}_t \cdot \mathbf{x}_t|$ is often interpreted as proportional to the *confidence* in the prediction. (In implementation we usually use the mean weight vector for making predictions.) When the true label $z_t \in \{-1, +1\}$ is revealed, CW algorithm updates the weight distribution by choosing a new weight distribution closest, in the KL divergence sense, to the current weight distribution $\mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$, while ensuring that the probability of a correct prediction for the current training example is no smaller than the confidence parameter $\eta \in (0.5, 1)$:

$$(\boldsymbol{\mu}_{t+1}, \Sigma_{t+1}) = \arg\min_{\boldsymbol{\mu},\Sigma} D_{KL}\big(\mathcal{N}(\boldsymbol{\mu}, \Sigma)||\mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)\big) \quad (1)$$

$$\text{s.t. } \Pr_{\mathbf{w}\sim\mathcal{N}(\boldsymbol{\mu},\Sigma)}[z_t\mathbf{w}\cdot\mathbf{x}_t \geq 0] \geq \eta. \quad (2)$$

The closed-form solution [Crammer *et al.*, 2008] to this constrained optimization problem is given by,

$$\begin{aligned}\boldsymbol{\mu}_{t+1} &= \boldsymbol{\mu}_t + \alpha_t z_t \Sigma_t \mathbf{x}_t \\ \Sigma_{t+1} &= \Sigma_t - \beta_t \Sigma_t \mathbf{x}_t \mathbf{x}_t^\top \Sigma_t\end{aligned}$$

with the updating coefficients

$$\begin{aligned}\alpha_t &= \max\left\{0, \frac{1}{v_t\xi}\left(-m_t\psi + \sqrt{m_t^2\frac{\phi^4}{4} + v_t\phi^2\xi}\right)\right\} \quad (3)\\ \beta_t &= \frac{\alpha_t\phi}{\sqrt{u_t} + v_t\alpha_t\phi}\end{aligned}$$

where $u_t = \frac{1}{4}\big(-\alpha_t v_t\phi + \sqrt{\alpha_t^2 v_t^2\phi^2 + 4v_t}\big)^2$, $v_t = \mathbf{x}_t^\top\Sigma_t\mathbf{x}_t$, $m_t = z_t\boldsymbol{\mu}_t\cdot\mathbf{x}_t$, $\phi = \Phi^{-1}(\eta)$ with Φ the cumulative distribution function of the standard normal distribution, $\psi = 1 + \frac{\phi^2}{2}$, and $\xi = 1 + \phi^2$.

Note that the constraint (2) can be rewritten in the form of an *adaptive, large-margin* constraint imposed by the current example (the RHS of (4)),

$$z_t\boldsymbol{\mu}\cdot\mathbf{x}_t \geq \phi\sqrt{\mathbf{x}_t^\top\Sigma\mathbf{x}_t}, \quad (4)$$

showing that CW is an *aggressive* algorithm, which updates its learning model not only in rounds with prediction mistakes (i.e., *conservative* algorithms) but also in rounds with margin violations, even if the predictions were correct. Furthermore, the adaptive margin constraint has been shown experimentally in CW and SCW to be an important property for achieving more effective and efficient online classification algorithms.

3.2 Adaptive Regularization of Weights (AROW)

The aggressive updating strategy employed by CW could lead to two contradicting effects. On the one hand, it results in the rapid learning effect by changing the weight distribution as much as necessary to satisfy the adaptive margin constraint (4). On the other hand, it could overfit when dealing with noisy labels or non-separable cases. AROW ([Crammer *et al.*, 2009b], see also NAROW [Orabona and Crammer, 2010]) modifies CW by introducing an adaptive regularization term in its objective function, to isolate and soften the impact of outliers. Specifically, it updates the learning model by solving the following unconstrained minimization problem:

$$\begin{aligned}(\boldsymbol{\mu}_{t+1}, \Sigma_{t+1}) = \arg\min_{\boldsymbol{\mu},\Sigma} &D_{KL}\big(\mathcal{N}(\boldsymbol{\mu}, \Sigma)||\mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)\big) \\ &+ \frac{1}{2\rho}\ell_{h^2}(z_t, \boldsymbol{\mu}\cdot\mathbf{x}_t) + \frac{1}{2\rho}\mathbf{x}_t^\top\Sigma\mathbf{x}_t, \quad (5)\end{aligned}$$

where $\rho > 0$ is a tradeoff parameter, and $\ell_{h^2}(z_t, \boldsymbol{\mu}\cdot\mathbf{x}_t) = \big(\max\{0, 1 - z_t\boldsymbol{\mu}\cdot\mathbf{x}_t\}\big)^2$ is the squared-hinge loss suffered on the current example $(\mathbf{x}_t, z_t)$ using the weight vector $\boldsymbol{\mu}$ for prediction. The desirable adaptive margin constraint in (4), however, is not imposed in this objective formulation anymore. The optimization problem in (5) has a closed-form solution similar to (3), but with simpler updating coefficients:

$$\alpha_t = \max\{0, 1 - z_t\boldsymbol{\mu}_t\cdot\mathbf{x}_t\}\beta_t; \beta_t = \frac{1}{v_t + \rho}.$$

3.3 Soft Confidence-Weighted Learning (SCW)

The Soft Confidence-Weighted (SCW) learning method [Wang *et al.*, 2012] has recently been proposed to address the above problems of CW and AROW, by adopting the soft-margin idea in Support Vector Machines (SVMs) (see also soft-margin Passive-Aggressive Algorithms [Crammer *et al.*, 2006]) to the case of CW learning method. Specifically, the SCW algorithm employs a parameter C to tradeoff the conservativeness and aggressiveness, and recasts the CW constraint as an adaptive regularizer in an unconstrained minimization problem on each round:

$$\begin{aligned}(\boldsymbol{\mu}_{t+1}, \Sigma_{t+1}) = \arg\min_{\boldsymbol{\mu},\Sigma} &D_{KL}\big(\mathcal{N}(\boldsymbol{\mu}, \Sigma)||\mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)\big) \\ &+ Cl^\phi\big(\mathcal{N}(\boldsymbol{\mu}, \Sigma); (\mathbf{x}_t, z_t)\big), \quad (6)\end{aligned}$$

where

$$l^{\phi}\big(\mathcal{N}(\boldsymbol{\mu},\Sigma);(\mathbf{x}_t,z_t)\big)=\max\left\{0,\phi\sqrt{\mathbf{x}_t^{\top}\Sigma\mathbf{x}_t}-z_t\boldsymbol{\mu}\cdot\mathbf{x}_t\right\}$$

is a confidence loss function for penalizing the violation of the adaptive margin constraint (4). The closed-form solution to this optimization problem is exactly the same as in CW algorithm, except that the updating coefficient α_t is slightly modified:

$$\alpha_t=\min\left\{C,\max\{0,\frac{1}{v_t\xi}(-m_t\psi+\sqrt{m_t^2\frac{\phi^4}{4}+v_t\phi^2\xi})\}\right\}.$$

4 Efficient Exploration for Multiclass Prediction with Binary Feedback

In this section, we present our general framework for dealing with multiclass prediction with binary feedback. In the next section, we will elaborate on implementation details of the algorithm.

Our algorithm maintains a set of M SCW binary classifiers $\{(\boldsymbol{\mu}^i,\Sigma^i)\}_{i=1}^M$ initialized with $(\boldsymbol{\mu}_0=\mathbf{0},\Sigma_0=\mathbf{I})$. Denote $\widehat{\Delta}_t^i=\boldsymbol{\mu}_t^i\cdot\mathbf{x}_t$ the prediction margin of classifier i on the current instance $\mathbf{x}_t$. The predicted label in each round is chosen based on the binary classifier having the largest prediction margin, i.e.,

$$\hat{y}_t=\arg\max_{i=1,\ldots,M}\big(\widehat{\Delta}_t^i\big).$$

The learner observes the binary feedback $z_t\in\{+1,-1\}$ indicating whether the prediction $\hat{y}_t\in\{1,...,M\}$ was correct (i.e., $\hat{y}_t=y_t$) or not (i.e., $\hat{y}_t\neq y_t$) when compared to the true label y_t.

4.1 Dealing with Binary Feedback

Consider a particular binary classifier i in our model. Since the weight vector $\mathbf{w}_t^i$ is drawn from the multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_t^i,\Sigma_t^i)$ over linear classifiers, its prediction margin is a random variable having a univariate Gaussian distribution, $\mathcal{N}\left(\widehat{\Delta}_t^i,v_t^i\right)$, where $v_t^i=\mathbf{x}_t^{\top}\Sigma_t^i\mathbf{x}_t$ as defined in section 3.1 acts as the variance of the prediction margin estimated by classifier i.

For the case of multiclass prediction with binary feedback, we follow the widely used approach of "optimism in the face of uncertainty" when dealing with the exploration-exploitation tradeoff, defining our *upper confidence bound* (UCB) of the prediction margin to be $\mathrm{UCB}_t^i=\widehat{\Delta}_t^i+k\sqrt{v_t^i}$, with $k\geq 0$ a tunable parameter to control the exploration-exploitation tradeoff. Algorithms 1 summarizes our proposed Upper Confidence Weighted Learning (UCWL) algorithm.

We want to justify this choice. Under the *realizability* assumption (i.e., the linear label noise model [Crammer and Gentile, 2011]), assume that the mean weight vector $\boldsymbol{\mu}_t^i$ converges to the optimal Bayes classifier $\boldsymbol{u}^i$. Then the probability, with respect to the *current* model distribution, that at any round t the Bayes optimal margin $\Delta_t^i=\boldsymbol{u}^{i\top}\mathbf{x}_t$, does *not exceed* the UCB is given by,

$$\Pr[\Delta_t^i\leq\mathrm{UCB}_t^i]=\Phi\left(\frac{\mathrm{UCB}_t^i-\widehat{\Delta}_t^i}{\sqrt{v_t^i}}\right)=\Phi(k),$$

with $\Phi(\cdot)$ denotes the cumulative function of the normal distribution, as mentioned before. For example, with the "two standard deviation" rule (i.e., $k=2$), we have $\Phi(2)\approx 0.9772$. Hence the UCB defined above can be used as an informative upper bound of the prediction margins given by the best binary classifiers chosen in hindsight (i.e., the Bayes optimal binary classifiers $\boldsymbol{u}^i$).

Algorithm 1: Upper Confidence Weighted Learning (UCWL). Parameters: $\eta\in(0.5,1);C,k>0$.

```
//Initialization
φ = Φ^{-1}(η), ψ = 1 + φ^2/2, ξ = 1 + φ^2.
for i = 1 : M do
    μ^i = 0 // d × 1 vector
    Σ^i = I // d × d matrix
end
//Main interactive learning loop
for t = 1, 2, ... do
    Receive new instance x_t ∈ R^d
    for i = 1 : M do
        //Calculate individual UCB margins
        UCB_t^i = μ^i · x_t + k sqrt(x_t^T Σ^i x_t)
    end
    //Select prediction with highest UCB
    Predict ŷ_t = arg max_{i=1,...,M} (UCB_t^i)
    Observe feedback z_t ∈ {+1, −1}
    Update selected classifier: //let † denote ŷ_t
    if z_t μ^† · x_t < φ sqrt(x_t^T Σ^† x_t) then
        //Constraint in Eq.(4) violated
        μ^† ← μ^† + α_t z_t Σ^† x_t
        Σ^† ← Σ^† − β_t Σ^† x_t x_t^T Σ^†
        //α_t, β_t as described in section 3.3
    end
end
```

5 Experiments

Here, we compare UCWL with other state-of-the-art algorithms on three different datasets, under different artificially generated incorrect feedback (noise) conditions. We report three performance metrics: online mistake rate (#mistakes / #predictions), runtime, and the number of model updates (i.e., the number of support vectors in the kernelized version). We conducted the experiments by first determining the best parameters for each method on each dataset, then applied each algorithm 20 times using these best parameters on each dataset, each time with a randomly permuted sequence (same

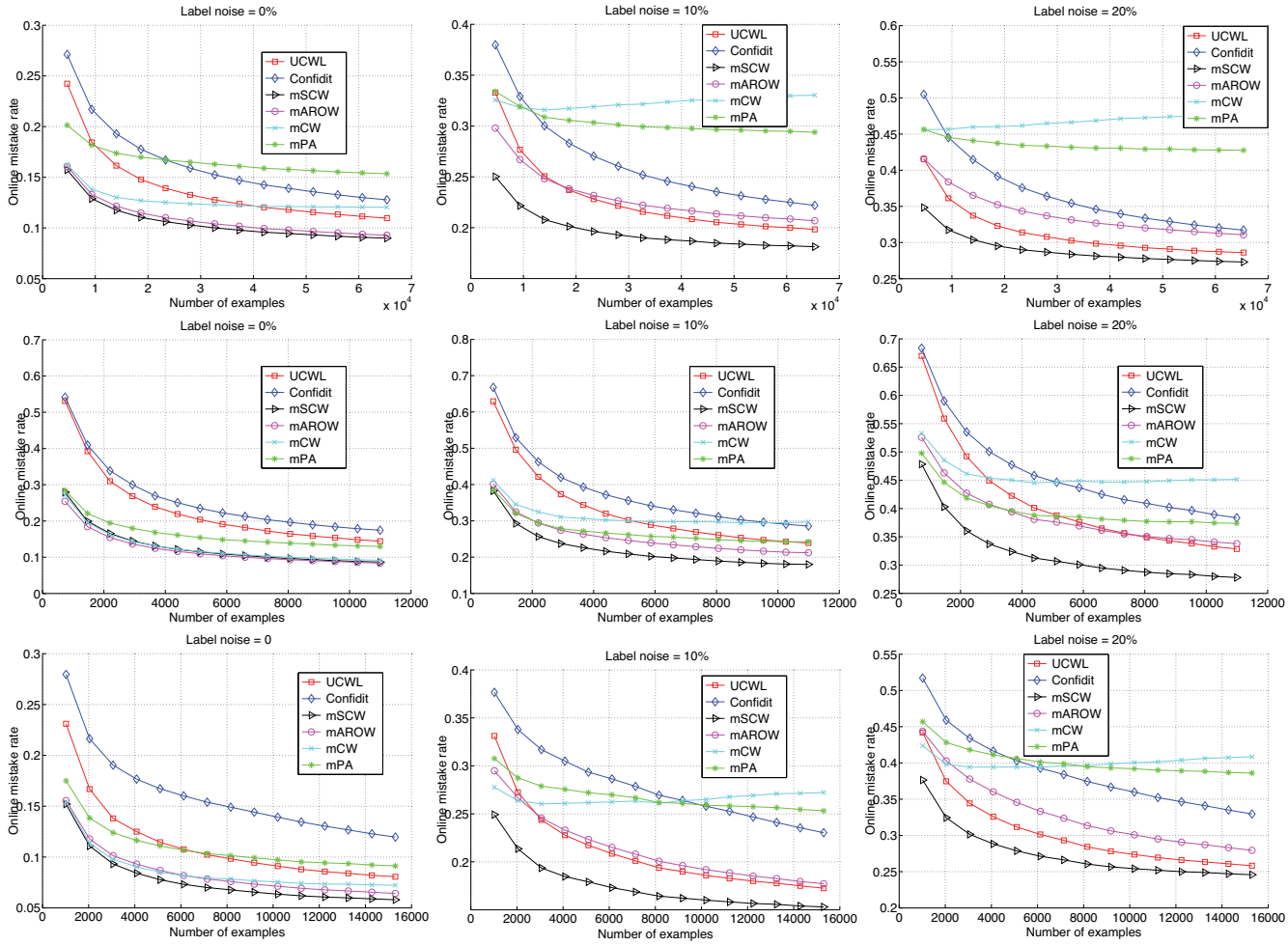

Figure 2: Online mistake rate for the MNIST dataset (top), USPS dataset (middle), and the GESTURE dataset (bottom) with label noise levels 0%, 10%, and 20%.

sequence for all algorithms). All the results reported are averaged over the 20 repetitions, and statistically significant.

Datasets. We used three multiclass datasets, two public and one internally collected. 1. MNIST, 2. USPS[2], and 3. IDSIA's hand gesture dataset (GESTURE) collected using a swarm of 13 footbots [Giusti *et al.*, 2012]). The MNIST and USPS datasets consist of grayscale images of ten handwritten digits from 0 to 9. MNIST has 70,000 examples of 10 classes, each instance has 784 attributes. USPS has 11,000 examples of 10 classes, each instance has 256 attributes. All feature values are normalized to $[0, 1]$ before training. Note that these features are raw pixel values, solely serving the purpose to evaluate *online learning* algorithms.

The GESTURE dataset [Giusti *et al.*, 2012] consists of 74,000 images, taken by a footbot, of a human's hand gesture and was acquired using onboard cameras from different viewpoints and distances. Since the main focus of the current paper is on classification with binary feedback, and not on prediction consensus within the swarm, we used 15,305 images from only 3 central viewpoints (-15, 0, and +15 degrees). There are 6 classes of gestures that the robots need to learn. Each image was preprocessed to segment the hand gesture, then a polygon was 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, a set of 110 features was extracted based on geometrical properties (e.g. shape and blob characteristics, image moments etc.) frequently used in the literature on similar tasks (see [Giusti *et al.*, 2012] for details).

Compared algorithms. Confidit and Banditron are the main competitors for UCWL. For Confidit, we used its deterministic ($\alpha = 1$), fully-diagonal simplified version, as noted in the original paper. Even though we compared with the Banditron [Kakade *et al.*, 2008], we do not report the results here since the results are very bad in comparison to the other algorithms.

We also want to see how well UCWL, which uses binary feedback, might compare with algorithms that use *full* feedback. *This is not a fair comparison*, and it would be a

[2]We used the 2 datasets from http://www.cs.nyu.edu/~roweis/data.html

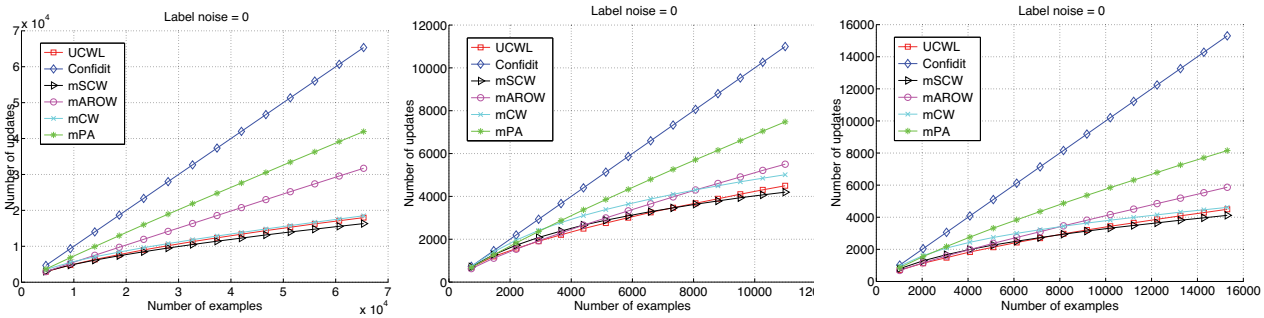

Figure 3: Number of updates for the MNIST dataset (left), USPS dataset (middle), and the GESTURE dataset (right) with label noise level 0%. Results for label noise levels 10% and 20% (not included) look almost identical.

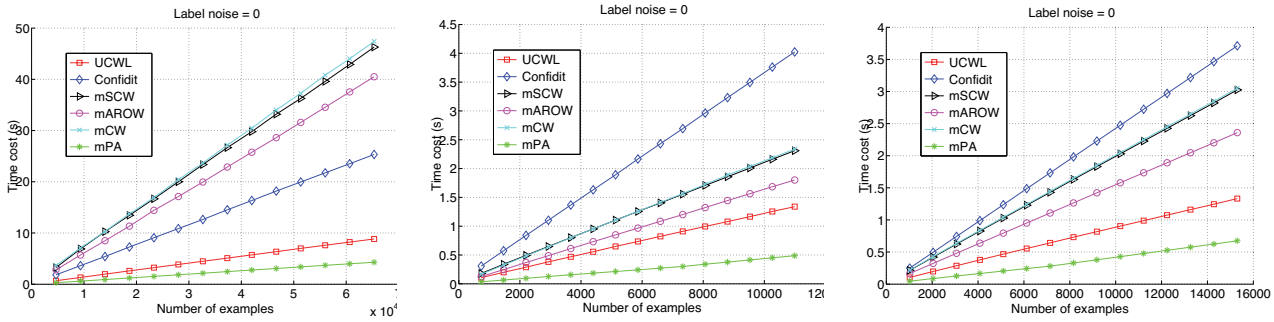

Figure 4: Computation time for the MNIST dataset (left), USPS dataset (middle), and the GESTURE dataset (right) with label noise level 0%. Results for label noise levels 10% and 20% (not included) look almost identical.

surprising result if UCWL can achieve better performance with much less information. To this end, we also compare with multiclass extensions of CW, SCW, AROW (second-order methods), and PA (first-order method). These are each state-of-the-art methods in their categories. PA has its multiclass version mPA already. For the second-order methods, we extend to their multiclass version using the general top-1 method described in [Crammer *et al.*, 2009a], which essentially updates two classifiers at each time: the classifier associated with the true label (promoted), and the incorrect classifier with highest margin (demoted).

Parameter setting. A single randomized run over each dataset is used to optimize the parameters for all algorithms in all experiments. Specifically, the slack parameter C in mPA, mSCW, UCWL was selected from the set $\{2^k\}_{k=-5}^{5}$ (same for the parameter r of mAROW), the confidence parameter η in mCW, mSCW, and UCWL was selected from the set $\{0.5+k*0.05\}_{k=1}^{9}$, and the UCB-exploration parameter k in Confidit and UCWL was selected from the set $\{0.2*k\}_{k=1}^{15}$.

Implementation details. Here, we note some details of UCWL specific to the experiments. First, we use the diagonal version for the covariance matrices $\{\Sigma^i\}_{i=1}^{M}$; which we found to reduce the computational complexity of the algorithm to be linear in the number of features, and also improved its accuracy as compared to the case using the full covariance matrices. This diagonalization step is also applied to mCW, mAROW, and mSCW. Second, when the feedback is "true", we have UCWL update the *selected* classifier only, instead of updating all the binary classifiers. Due to this, the performance of the algorithm was further improved. Thus in the pseudocode of Algorithm 1 we used this update method: only the selected classifier indexed by $\hat{y}_t$ (and denoted by †), i.e., the pair $(\boldsymbol{\mu}^\dagger, \Sigma^\dagger)$, is updated.

Results. Comparison results[3] with different label noise levels (10%, 20%, and 30%) are shown in Figures 2- 4. Fig.2 reports the online mistake rates for each algorithm. Crucially, UCWL beats Confidit and Banditron (not reported due to poor results) in every situation tested. Again, these are the only algorithms tested that use binary feedback, and UCWL delivers the top performance. Now, in the original Confidit paper, an interesting result was reported: Confidit, while working with binary feedback, showed comparable performance to full feedback first order methods, such as mPA. We also observe this in our results, and further, we see that in noisy label settings the binary-feedback using UCWL can beat the *second-order* methods mAROW, and mCW, which, again, use *full* feedback. Only the powerful full-feedback second-order mSCW beats UCWL consistently. Though of course mistake rate suffers with noise, the effectiveness of UCWL increases as the noise increases.

Fig. 3 shows the number of updates for each algorithm

[3]Since there are no significant differences in the results between different variants of the baseline algorithms, here we just report results, for mSCW and mPA, using their type-I variant, and for mCW and mAROW, their full-variance variant followed by a diagonalization step.

on the data sequences, while Fig. 4 shows the computation time spent. We see that UCWL has among the fewest number of updates due to the aforementioned four salient properties, especially the adaptive, large-margin constraint. Further, UCWL makes an update (which constitutes the main computation) only for the selected classifier. As a result, UCWL is much faster than all second-order algorithms, and almost as fast as the fastest, first-order algorithm mPA. The number of updates for mSCW is also low, but this is slightly misleading — mSCW must update two classifiers each time (promote/demote top-1 style [Crammer *et al.*, 2009a]), but it has to update *less often*, since, given full information, it is more often correct (and it will not update when the margin constraint (4) is satisfied). Since mSCW updates both classifiers, its speed is significantly slower than UCWL's.

Again, we want to note that the only *fair* comparison is between UCWL and Confidit, and UCWL clearly emerges the better in our experiments. It is surprising that UCWL can perform so well under noisy label conditions with binary feedback, when other algorithms get full feedback on the same data.

6 Conclusion

We introduced the Upper Confidence Weighted Learning algorithm for exploration in the online learning with binary feedback setting. UCWL extends the Soft Confidence Weighted online learning framework to deal with binary feedback using the Upper Confidence Bound framework. This novel formulation suits human-robot interaction and led to state of the art performance in terms of accuracy, computational efficiency, and robustness against label noise.

Acknowledgements. We would like to thank Jawad Nagi for providing us with the GESTURE dataset.

References

[Auer, 2003] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2003.

[Azoury and Warmuth, 2001] K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Journal of Machine Learning Research*, 43(3):211–246, Jun. 2001.

[Cesa-Bianchi and Lugosi, 2006] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[Cesa-Bianchi *et al.*, 2005] N. Cesa-Bianchi, A. Conconi, and C. Gentile. A second-order perceptron algorithm. *SIAM Journal on Computing*, 34(3):640–668, 2005.

[Crammer and Gentile, 2011] K. Crammer and C. Gentile. Multiclass classification with bandit feedback using adaptive regularization. In *Proceedings of the International Conference on Machine Learning*, pages 273–280, 2011.

[Crammer *et al.*, 2006] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

[Crammer *et al.*, 2008] K. Crammer, M.D. Fern, and O. Pereira. Exact convex confidence-weighted learning. In *Advances in Neural Information Processing Systems 22*, pages 345–352, 2008.

[Crammer *et al.*, 2009a] K. Crammer, M. Dredze, and A. Kulesza. Multi-class confidence weighted algorithms. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2*, pages 496–504, 2009.

[Crammer *et al.*, 2009b] K. Crammer, A. Kulesza, and M. Dredze. Adaptive regularization of weight vectors. In *Advances in Neural Information Processing Systems*, volume 22, pages 414–422, 2009.

[Giusti *et al.*, 2012] 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)*, pages 551–558, 2012.

[Kakade *et al.*, 2008] S.M. Kakade, S. Shalev-Shwartz, and A. Tewari. Efficient bandit algorithms for online multiclass prediction. In *Proceedings of the International Conference on Machine Learning*, pages 440–447, 2008.

[Orabona and Crammer, 2010] F. Orabona and K. Crammer. New adaptive algorithms for online classification. In *Advances in Neural Information Processing Systems*, pages 1840–1848, 2010.

[Schmidhuber, 2010] J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990-2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

[Szepesvári, 2010] C. Szepesvári. Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 4(1):1–103, 2010.

[Vovk, 2001] V. Vovk. Competitive on-line statistics. *International Statistical Review*, 69(2):213–248, 2001.

[Wang *et al.*, 2012] J. Wang, P. Zhao, and S.C.H. Hoi. Exact soft confidence-weighted learning. In *Proceedings of the International Conference on Machine Learning*, 2012.