Human Boosting

Harsh Pareek HARSHP@CS.UTEXAS.EDU
Pradeep Ravikumar PRADEEPR@CS.UTEXAS.EDU
Department of Computer Science, The University of Texas, Austin, TX 78712, USA

Abstract

Humans may be exceptional learners but they have biological limitations and moreover, inductive biases similar to machine learning algorithms. This puts limits on human learning ability and on the kinds of learning tasks humans can easily handle. In this paper, we consider the problem of "boosting" human learners to extend the learning ability of human learners and achieve improved performance on tasks which individual humans find difficult. We consider classification (category learning) tasks, propose a boosting algorithm for human learners and give theoretical justifications. We conduct experiments using Amazon's Mechanical Turk on two synthetic datasets – a crosshair task with a nonlinear decision boundary and a gabor patch task with a linear boundary but which is inaccessible to human learners – and one real world dataset – the Opinion Spam detection task introduced in (Ott et al., 2011). Our results show that boosting human learners produces gains in accuracy and can overcome some fundamental limitations of human learners.

1. What does it mean to boost humans?

Human learning ability is indeed a piece of work, but man is certainly not infinite in faculties. There has been a long line of work on human learning in the psychology and cognitive science literature. The study of human category learning in particular — studying how humans classify objects into different categories — has seen considerable advances in recent years (Ashby & Maddox, 2005). Humans have been shown to have differing learning abilities depending on the nature of the task at hand. An important distinction for instance can be made between "rule-based" and "information-integration" category learning tasks, depending on the type of decision boundary. Rule-based tasks have easily verbalizable classification rules, and typically depend on a single semantic dimension (e.g. "the width of the lines is small"), or simple logical operations on a few such dimensions (e.g. "the width of the lines is small, and the color is red"). In information-integration tasks on the other hand, different semantic dimensions are combined at a *predecisional* stage and the learner is unable to precisely describe the classification rule they are using. An example of an information-integration task is object recognition, which humans can perform quickly using little to no working memory and attention, but the strategy for which is hard to describe in words. Research over the last two decades has led to the understanding(Ashby & Maddox, 2011) that learning in humans is mediated by multiple systems, each governed by separate specialized biological processes. Deficits in any of these processes could lead to deficits in the corresponding aspects of human learning. Another interesting limit to human category learning arises from the mismatch between the discriminative dimensions in the data and those dimensions that humans find natural. It has been observed that some pairs of perceptual dimensions are "separable" for humans, so that they are able to reason about one dimension without involving the other; for instance, humans can separately reason about the shape and the color of an object. On the other hand, there are dimension pairs that are "integral", which humans interpret holistically and they find it difficult to reason about any of the individual dimensions separately. For example, for describing a color, red, green and blue are separable dimensions while hue and saturation are integral dimensions. These fundamental limitations, and practical considerations we discuss in Section 2 limit human learning ability.

In the face of seemingly intrinsic limits of human category learning, here we ask whether it is possible to

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

"boost" the learning ability of humans? This question might seem ill-posed, but precisely such a question was posed in the context of computational learning theory by Kearns & Valiant (1989), where they asked whether a "weak learner", specifically a learning algorithm which can learn a classifier that is only slightly better than random guessing, could be "boosted" into an arbitrarily accurate strong learning algorithm?. Interestingly, Schapire (1990) was able to answer this question in the affirmative. This technique, called boosting, has been the subject of considerable research over the last two decades, and its performance is now reasonably well-understood (Friedman et al., 2000; Collins et al., 2002; Meir & Rätsch, 2003). Boosting constructs an ensemble of learners by providing a weak learning algorithm with iteratively modified training datasets, putting increasingly greater weights on the examples deemed more difficult as iterations proceed. We review boosting in Section 2.

In this paper, we investigate the possibility of using humans as weak learners and boosting them to strongly learn complex concepts better than any single human. Our principal contributions are as follows. (a) The standard boosting algorithms are *not amenable* to human learners (weighted examples may not be appropriately interpreted by humans for instance). We describe the modifications that need to be made, along with theoretical justifications, that extend the state of the art in the analysis of boosting. (b) We design suitable experiments for boosting on humans: we consider two synthetic tasks, which we call the crosshair and gabor tasks, and one real world task, detecting fake reviews in the Opinion Spam dataset of (Ott et al., 2011). (c) We run category learning experiments on the crowdsourcing platform, Amazon's Mechanical Turk. Our experimental results show that boosting can overcome some intrinsic limitations of human learners and produce gains in accuracy.

Related work: Our work is in the vein of interesting recent work by (Zhu et al., 2011) which has focused on combining human classifiers using co-training. It will be useful to position our work against the frameworks of Human Computation and Crowdsourcing. Human Computation (Quinn & Bederson, 2011) has been termed "a paradigm for utilizing human processing power to solve problems that computers cannot yet solve". Thus, it uses humans as "subroutines" for solving sub-tasks, particularly those that computers find difficult. Crowdsourcing on the other hand, has been termed "the outsourcing of a job typically performed by an expert to a crowd that typically have lower-expertise but are large in number". The key idea in crowdsourcing is to suitably aggregate the outputs of the members of the crowd and this can at times rival the performance of an expert. A key contrast of our work with human computation and crowdsourcing, is that the goal in human boosting is to use humans not as fixed classifiers and subroutines, but as *changing and learning algorithms*. Moreover, as we detail in the main section of the paper, the protocol of using humans in our human boosting method, is complicated and even asynchronous, unlike typical tasks in crowdsourcing. Nonetheless, the work here is very related to these frameworks. As such, our research raises the question of whether a richer use of humans (e.g. as learning algorithms) would allow them to be embedded in non-trivial ways within other larger machine learning frameworks? Our work can also be positioned as the converse of active learning. In active learning, eg.(Settles, 2010), the human acts as an oracle and the algorithm asks the human to label what it believes are the "most informative" examples: a machine performs classification and is guided by a human. In our work, the human performs the classification task and the machine guides humans.

Applications: The question of whether human learners can be boosted is certainly interesting from a cognitive science point of view. But as we detail, it has a number of practical applications as well.

The first is the use of humans as "supervised" feature extractors. For many natural tasks such as those in vision and text, humans have access to evolutionarily evolved implicit feature spaces, which we find difficult to even capture with a computer. If each human learner is asked to learn a "decision stump" involving a single or a few features of their choosing within our boosting framework, then we can use these to extract task-driven implicit human features. It is useful to compare this to manifold learning. For instance, in the high dimensional space of images represented as a vector of their pixels, images of a given face lie on a manifold, and face recognition essentially requires a learner to identify this manifold. However, for any manifold, there is an appropriate mapping into lower-dimensional Euclidean spaces where classification becomes easy. Manifold learning techniques aim to extract these embedding using large amounts of data. Extracting implicit features from humans could potentially allow us to achieve similar performance using much smaller amounts of training data.

A human boosting framework can also be used to bootstrap labeled data for difficult learning problems. Using a small number of labeled examples, we could have humans "strongly learn" complex concepts and create larger datasets for consumption by machine algo-

rithms. Indeed, ground truth labels for typical machine learning datasets are either provided by or validated by humans. Thus, the problems selected are typically those in which humans excel. Boosting human learners allows us to solve problems which even humans find difficult, and expands the frontier of machine learning research. These methods could be used to replace a domain expert or a complexly engineered solution with a number of non-expert humans. While it is true that it is very expensive to use humans, the success of crowdsourcing has shown that for several problems, it is cheaper on the whole to use humans than to employ engineering solutions.

2. Boosting Human Learners

We start the section by discussing the characteristics of humans as "learners," in the context of machine learning. We then discuss boosting, and Adaboost, and why modifications to the Adaboost are essential to be amenable to the characteristics of human learners. We then detail our "HumanBoost" algorithm. Finally, we provide a convergence analysis for an idealized version of our HumanBoost algorithm.

Human Learners: Humans as learners differ from machine learning algorithms in several ways. First, human learners are a black box, in the sense that we do not have access to their inner workings. Second, machine learners use high dimensional feature vectors as input, which humans cannot interpret or visualize. On the other hand, humans do understand images, video, audio and text, and to an extent that even state of the art machine learning algorithms cannot capture. In this paper in particular, we will *represent* input vectors via intuitive visual stimuli for use with human learners. Third, machine learning algorithms take large numbers of training examples as input. Human learners can only be given a small number of training examples. Finally, it is not clear whether humans can handle classification noise and how they would respond to differing distributions in the data. There is some evidence (Fried & Holyoak, 1984) that humans can learn distributions and can learn both discriminative and generative models (Hsu et al., 2010).

Boosting and Adaboost: The framework of boosting is now understood(Friedman et al., 2000) to greedily fit an additive model of the form $H(x) = \sum_t \alpha_t h_t(x)$ to a classification-surrogate loss such as the exponential function; and where the functions $\{h_t(\cdot)\}$ are the classifiers learnt by a "weak" learning algorithm, and $\{\alpha_t\}$ are the weights assigned to these classifiers. With a careful selection of weights and modified training sets for the intermediate steps, it can be shown that the output classifier $H(x)$ is a "strong" classifier that might well have been learnt by a "strong" learning algorithm.

We first describe the popular boosting algorithm of Adaboost (Schapire, 2003) as it is closely related to our proposed algorithm. In each round, Adaboost maintains a weighted distribution over the training examples, with weights corresponding to the difficulty of that example. Initially, all examples are weighted equally. Each iteration of Adaboost has three steps: (a) train the weak-learning algorithm on this weighted data distribution, (b) compute its classification error given the new data distribution (c) multiplicatively increase the weights on the "difficult" examples. This ensures that the weak learning algorithm tries to fit those points more accurately in the next iteration. Indeed, over the past decade, Adaboost has been analyzed both experimentally (Dietterich, 2000) and theoretically (Friedman et al., 2000; Collins et al., 2002).

Certain properties of Adaboost make it amenable for use with human learners. First, it is a wrapper procedure, i.e. it does not need to know the inner workings of the weak learner; this makes it conducive for use with "black box" human learners. Second, it does not directly operate on the training data (x_i, y_i) (where x_i is the input stimulus and y_i is the label), so x_i could be complex stimuli such as text or video. Third, there is variability among humans learners, they are not reliable and their output is noisy. Adaboost addresses this by adaptively reweighting the bad human learners.

Certain other parts of Adaboost need to be modified. Adaboost reweights training examples. Humans may not interpret these weights correctly or consistently, and simply presenting the numerical weights would be unreliable. The theoretical guarantees of Adaboost fail in such a case. One potential solution is to resample training examples, drawn from a multinomial distribution corresponding to Adaboost weights. The caveat with this is that it will lead to examples with large weights being repeated, and humans may not interpret repetitions correctly. For instance, if we consider stimuli such as text reviews (as in section 3.2.3), it will be obvious to the learner if a review was repeated, and the learned results would be unreliable.

Proposed algorithm: Our proposed boosting algorithm, called HumanBoost, is detailed in Algorithm 1. Recall that any boosting algorithm has the following three steps in each iteration: (a) train a weak-learner on the new data distribution, (b) compute the classification error given the new data distribution, and (c) create a new data distribution in light of the difficult examples in the previous round. We now discuss how

Algorithm 1 HumanBoost

Input: MTurk oracle $\mathcal{O}$, $D = \{(x_i, y_i)\}_{i=1}^N$, δ_t,ϵ, m_t
Set weights $w_i^1 = 1/N$, $i = 1, \ldots N$
for $t = 1, \ldots, T$ **do**
  Sample training set S using FILTER(m_t, w^t, δ_t).
  Use $\mathcal{O}$ to obtain weak human classifier h_t for S
  Label the test set $T = D - S$ using h_t.
  Edge $\gamma_t = \sum_{i \in T} \mathbf{1}[y_i = h(x_i)] w_i^t / \sum_{i \in T} w_i^t - 0.5$.
  Set weight $\alpha_t \leftarrow 0.5 \log(0.5 + \gamma_t)/(0.5 - \gamma_t)$
  Update $H_t(x) \leftarrow H_{t-1}(x) + h_t(x)$.
  Update $w_i^{t+1} \leftarrow 1/(1 + \exp(y_i H_t(x_i)))$.
end for
RETURN strong learner $H_T(x) = \text{sign}(\sum_t \alpha_t h_t(x))$.

function FILTER(m_t, w^t, δ_t)
  $r =$#calls to filter; $\delta' = \delta_t / r(r+1)$; $i_{\max} = 2\delta_t/\epsilon$.
  Permute (D, w); Set $S = \phi$; $i = 1$.
  while $|S| \leq m_t$ **do**
    $(x, y) \leftarrow D(i)$; $i \leftarrow i + 1$.
    With probability w_i^t: add (x, y) to S.
    If no update to S in past $i_{\max}$ iters, hypothesis H_t is sufficiently accurate; exit program.
  end while
  RETURN S.
end function

the HumanBoost algorithm performs these three key steps. HumanBoost maintains the new distribution via weights over the data points, just as in Adaboost. But for training a human weak-learner, we do not want to provide either weights over the samples, or a resampled data set with potentially repeated samples. Instead, we perform sampling without replacement, with probability proportional to the normalized weights. Such subsampling without replacement in the context of boosting was first experimentally analyzed for boosted trees in (Friedman, 2002) in the context of Stochastic Gradient Boosting. Here, we use a rejection sampling framework similar to FilterBoost(Bradley & Schapire, 2008) and that of (Zadrozny et al., 2003), and as described in the function FILTER in algorithm 1. We go through the samples in a random order, and reject easy examples one by one till we collect a sufficient number of difficult examples S. For step (b), and computing the weights α_t, we need to measure the weighted accuracy of h_t. As we detail in the experimental section, we train the human learner h_t over S in an online fashion, and then ask them to label examples from $D - S$. Learners may memorize examples from S so we cannot use S in the test set. This introduces an unavoidable error in the computation of α_t, as S contains the most difficult examples and so $D - S$ contains easy examples. However, for human learners, S is small, and if the learner has high accuracy on $D - S$, we can assume that it is sufficiently accurate. A key algorithmic design aspect of HumanBoost as pertaining to step (b) is the classification-surrogate loss used to compute the classification error, and the weights. While Adaboost uses the exponential loss, its weights can grow arbitrarily large which reduces its tolerance to noise. One robust approach as in Madaboost(Domingo & Watanabe, 2000), is to truncate the weights explicitly, to a maximum of 1. We use the logistic function as in LogitBoost(Friedman et al., 2000). Step (c) of creating the new data distribution is similar to that of Adaboost, given the classification error in Step (b).

Remark: Algorithm 1 extends Discrete Adaboost using a logistic weight function and rejection sampling without replacement. Note that we can also extend "Real Adaboost" in a similar fashion. However, Real Adaboost requires classifiers to output probability estimates. Experiments have shown that humans are not good at estimating probabilities(Cosmides & Tooby, 1996). Teaching humans to output confidence values or calibrating their output for Real Adaboost is an interesting direction for future research. Another extension of Adaboost potentially amenable to humans is the boosting by relabeling, also known as Agnostic Boosting(Ben-David et al., 2001). In agnostic boosting, easy examples are randomly relabeled so that the weak learner can focus on more difficult examples. This effectively adds label noise to the dataset proportional to the weight in the distribution. Whether humans will interpret this noise correctly is an open question, which we will investigate in future work.

Sampling: We can show that our sampling procedure is efficient, following the lines of (Bradley & Schapire, 2008). Specifically, if the rejective filtering procedure terminates without giving enough examples, then the error err_t of our hypothesis H is small.

Proposition 2.1 *If in any call of* FILTER, $n \geq \frac{2}{\epsilon} \ln(1/\delta)$ *consecutive examples are rejected, then* $err_t \leq \epsilon$ *with probability* $1 - \delta$.

Analysis: A popular class of analytical results in boosting (Collins et al., 2002; Bradley & Schapire, 2008) use the recurrences arising from the new data distribution reweighting difficult examples in the previous iteration, and the weak-learners having an "edge" over random guessing even on the new data distributions. Another recent line of analytical results are based on the view of boosting as greedy stagewise functional gradient descent (Mason et al., 1999; Friedman et al., 2000; Grubb & Bagnell, 2011). Recently, Grubb & Bagnell (2011) provided an elegant analysis of boosting under the restriction of using weak learn-

ers, by interpreting it as *restricted* gradient descent in function space. Our analysis is an extension of this recent result, where we interpret our algorithm as a *stochastic* version of their restricted gradient descent.

Our presentation here is terse for lack of space, and we refer to (Grubb & Bagnell, 2011) for extended details of their framework. Consider the function space $L^2(\mathcal{X}, \mathbb{R}, P)$ of square-integrable functions, equipped with the inner product $\langle f, g\rangle_P = \mathbb{E}_P[f(x)g(x)] = \int_{\mathcal{X}} f(x)g(x)dP(x)$. Denote by $\hat{P}$ the empirical distribution over the data $\{(x_i, y_i)_{i=1}^N\}$, and the corresponding inner product, $\langle f, g\rangle_{\hat{P}} = \sum_{i=1}^N f(x_i)g(x_i)$. Consider the objective $\mathcal{R} : L^2(\hat{P}) \to \mathbb{R}$ derived from the empirical risk $\mathcal{R}_{emp}[f] = \frac{1}{N}\sum_{i=1}^N l(y_i f(x_i))$; assume the loss function l is convex, and differentiable (note that the function l in our case is the logistic function $l(t) = \log(1 + \exp(-t))$). Denote the "restriction set" of weak learners as H. Boosting can then be derived as a method to minimize $\mathcal{R}_{emp}[f]$ subject to the constraint that $f = \sum_t \alpha_t h_t$ with $h_t \in H$ and $\alpha_t \in \mathbb{R}$.

The subgradient of $\mathcal{R}_{emp}[f]$ at any f can be written as $\nabla\mathcal{R}_{emp}[f] = \{g | g(x_i) = \frac{\partial l(y_i f(x_i))}{\partial f(x)} = l'(y_i f(x_i))y_i\}$; denote this by ∇. The key insight in Grubb & Bagnell (2011) was to leverage the fact that each boosting step can be written as $f_{t+1} = f_t + \alpha_t h_t$, where $h_t \in \arg\max_{h\in H}\langle \nabla, h\rangle/\|h\|$, and $\alpha_t = -1/\eta_t\langle \nabla, h_t\rangle/\|h_t\|^2$, for some step-size η_t. Another key idea there was that the weak-learning assumption can be asserted as an approximate gradient condition: $\arg\max_{h\in H}\langle \nabla, h\rangle/\|h\| \geq \gamma\|\nabla\|$, for some $\gamma > 0$. They then relate this γ to the edge parameter used in typical boosting analyses. Note that the boosting step is in effect a *restricted* gradient descent, which they then leverage to show that (i) the algorithm asymptotically converges to the optimum of the desired loss function, and that (ii) it does so efficiently, i.e. at a rate polynomial in $1/\epsilon$, where ϵ is the desired error.

The key difference in our boosting algorithm when compared to the setting of (Grubb & Bagnell, 2011) is that we *sample* training data without replacement, and provide them to the learner without weights, so that instead of the ∇ derived above, we use a *stochastic* gradient ∇': where $\nabla'(x_i) = S_i y_i$, with $S_i \in \{0, 1\}$ is an indicator for data instance i, i.e. 0 if $x_i \notin S$ and 1 if $x_i \in S$. Our algorithm is thus a stochastic restricted gradient descent, and we can extend the analysis of (Grubb & Bagnell, 2011) to obtain convergence rates even in our setting. We first note that the function FILTER in 1 ensures that $p(x_i \in S) = 1/(1 + \exp(-y_i f(x_i))) = l'(y_i f(x_i))$ where f is the current hypothesis. It thus follows that the inclusion probability of x_i satisfies: $\mathbb{E}[S_i] = l'(y_i f(x_i))$, so that $\mathbb{E}[\nabla'] = \nabla$ on x_i. Our boosting algorithm can then be abstracted as performing the step: $f_{t+1} = f_t + \alpha_t h_t$, where $h_t \in \arg\max_{h\in H}\langle \nabla', h\rangle/\|h\|$, where ∇' is the stochastic gradient detailed above. For the purpose of analysis, set the weight as $\alpha_t = -1/\eta_t\langle \nabla', h_t\rangle/\|h_t\|^2$. Further, we require that the restriction set satisfy the weak-learning assumption with respect to the stochastic gradient, so that $\arg\max_{h\in H}\langle \nabla', h\rangle/\|h\| \geq \gamma\|\nabla'\|$, for some $\gamma > 0$. We can then extend the analysis of (Grubb & Bagnell, 2011) to obtain the following:

Theorem 2.2 *Let $\mathcal{R}_{emp}[f]$ be a α-strongly convex, and β-strongly smooth functional over $\mathcal{F} := L^2(\mathcal{X}, \mathbb{R}, \hat{P})$. Let $\mathcal{H} \subset \mathcal{F}$ be a restriction set with edge γ. Suppose further that the variance of the stochastic gradients is bounded so that $\mathbb{E}[\|\nabla'_t - \nabla_t\|] \leq \kappa\|\nabla_t\|$. Let $f^* \in \arg\min_{f\in\mathcal{F}}\mathcal{R}_{emp}[f]$. Given a starting point f_0, and step-size $\eta_t = 1/\beta$, after T iterations of our algorithm, we have:*

$$\mathbb{E}[\mathcal{R}_{emp}[f_T]] - \mathcal{R}_{emp}[f^*] \leq \rho^T\left[\mathcal{R}_{emp}[f_o]] - \mathcal{R}_{emp}[f^*]\right],$$

where $\rho := (1 - (2\alpha/\beta)(\gamma/4 - \kappa^2 - \kappa(1 + \gamma/2)))$, and where the expectation is over the randomization in our sampling procedure.

3. Experiments and Results

We first describe our experimental setup and then present our experimental results demonstrating Human Boosting on category learning tasks, where learners are expected to learn to distinguish between two classes A and B of stimuli, based on examples.

3.1. Experimental Setup

We conduct experiments using Amazon's Mechanical Turk(MTurk). MTurk is a crowdsourcing marketplace which connects *requesters*, who assign *HITs* (Human Intelligence Tasks), to *workers* who perform the HITs. MTurk is ideally suited to microtasks which can be performed quickly. Our program posted HITs on MTurk to recruit workers and led them to our server for experiments. On completing the experiment, they were given a unique random string to enter on MTurk as verification for payment. Since HITs are paid tasks, workers have the incentive to spam tasks, i.e. click through them quickly giving wrong results. Workers on MTurk are also typically less attentive than volunteers in laboratory experiments. To ensure worker attentiveness and to discourage spammers, a one second delay was given after every training example and a half second delay was given after every test example. Our experiments used workers with > 1000 approved HITs, and a $> 95\%$ acceptance rate. All HITs were designed to be completed in less than 20 minutes. Each

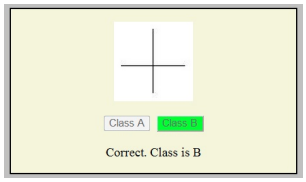


Figure 1. Interface presented to learners, with both visual and text feedback: correct answers are colored green, and displayed as text; incorrect answers are colored red.

HIT corresponds to one round of boosting and can only be picked up by a single worker, and we ensured that the workers in a run of the boosting algorithm were all distinct.

Interface: Our interface (Figure 1) is a point and click web-page built using python CGI and javascript. Each HIT is a session consisting of two phases, the training phase and the testing phase. In the training phase, the learner is shown a stimulus and presented two buttons, one for each class ("Class A" and "Class B"). After he chooses an option, he is told whether his response was correct, and the next stimulus is shown. Previous stimuli are not removed from the screen. This kind of training is called "Training with feedback", and is known to be superior to observational training where the learner is only shown examples from each class(Ashby et al., 2002). After the training phase, we ask the participant to decide on a rule, write it down and *verify* that this rule works on the training examples *they* have been provided. In the testing phase, workers are asked to use the learned rule to label unseen examples. Note that since all test examples are shown to each learner, the test examples will not have the same distribution as the reweighted training data (in fact, their distribution is the same as the original data). Human learners respond to the test examples as if they were unlabeled examples and this may cause drifting of the classification boundary (Zhu et al. (2010) call this the *test-item effect*). We attempt to mitigate this by asking them to explicitly write down the rule before they begin labeling examples.

3.2. Results

We present experiments on three stimuli: two synthetic, Crosshairs and Gabor patches, and one real world, the Opinion Spam detection dataset (Ott et al., 2011). For each stimulus, we contrast the performance of two groups of learners: the *control* group and the *boosted* group. The control group is trained on a small sample of the dataset and their accuracy on a test set is measured. Each learner in the boosted group is used only in one round of boosting. Humans in both groups are given the same number of training examples. The exact instructions given to the workers is included in supplementary materials. Synthetic stimuli prevent learners from using preconceived notions of classes, so that we can observe the effects of learning more clearly. Preconceived notions are unavoidable when using real world stimuli, however we mitigate this by hiding the true class labels and telling the learner only that there are two classes which need to be distinguished.

We compare the accuracy of the best learner in the control group to the boosted learner. In the case of noisy learners such as humans on Turk, the best learner performs better than the majority vote learner, and we just compare the performance of the best human learner with that of the boosted algorithm. Note that our algorithm is in fact complementary to crowdsourcing algorithms used to aggregate human learners such as those of (Dawid & Skene, 1979; Sheng et al., 2008), and we are solving the essentially different problem of boosting human learners.

3.2.1. Crosshair Task

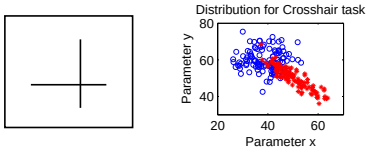
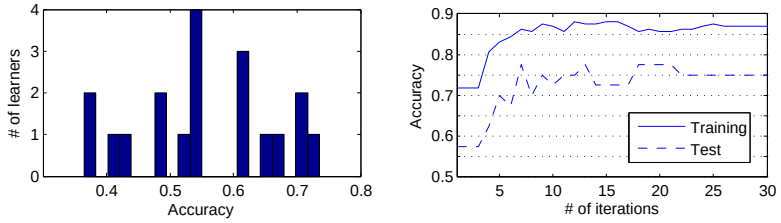


Figure 4. Crosshair Learning Task

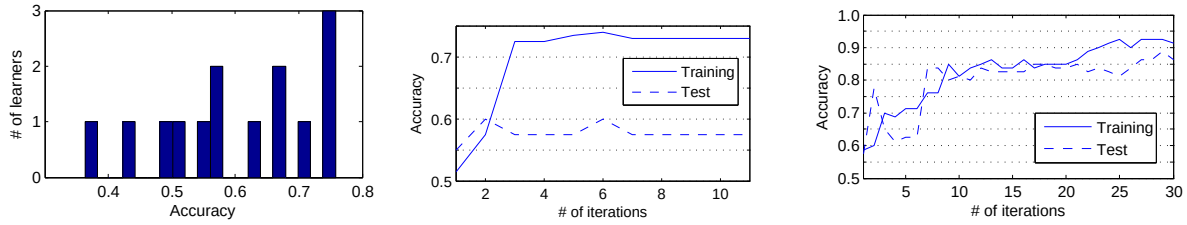
Two intersecting perpendicular lines, see Figure 4(a), are presented to the subject as a 100×100 pixel image. The class of the cross-hair image depends on the control dimensions, viz the x and y coordinate of the point of intersection. We draw classes from the overlapping gaussians in Figure 4(b). Preliminary experiments indicated that if the classes were separated, humans would easily learn a linear boundary involving both dimensions, but they report confusion if the classes overlap. Further, the optimal decision boundary in Figure 4(b) is quadratic and thus nonlinear. Since knowing the control parameters for a data point only probabilistically determine a class, this is an example of a probabilistic decision boundary as described in (Ashby & Maddox, 2005). The human learners were informed that some class labels may be incorrect. Due to the class overlap, the maximum accuracy achievable in this task is 85%. The aim of this experiment is to demonstrate the boosting can help recover from noisy input which confuses human learners.

Figure 2(a) shows a histogram of the human learners



(a) Histogram for control group (b) Boosting accuracy

Figure 2. Boosting on the Crosshair Task: The training accuracy of the boosted learners approaches the maximum possible accuracy of 85% in 10 rounds, performing better than the best control group learner. Boosting can overcome noise in the dataset



(a) Histogram for control group (b) Boosting with 20 examples (c) Boosting with 5 examples

Figure 3. Boosting on the Gabor Task: Given 20 examples in each round, the boosted learner suffers the same limitation as the best control group learner (uses only the stronger perceptual dimension). However, with 5 examples, boosting forces learners to concentrate on the discriminative features and achieves $> 90\%$ training accuracy in 30 rounds

in the control group who were trained with 5 examples each. They achieved an average test accuracy of $55.74\% \pm 11.35\%$ with a maximum of 70%. The reported rules were typically noisy decision stumps in either dimension. With 5 examples, humans are weak learners on this task. With more examples, humans do perform better, but report confusion about the class boundary. We perform experiments with 5 examples to demonstrate the effect of boosting.

Figure 2(b) shows the performance of the boosted learners. In 10 rounds, human boosting achieves 85% training accuracy and 75% test accuracy, which is better than the best control group learner. The gap between the training and test accuracy is called the "generalization error" and is a common occurrence in machine learning algorithms. This error typically goes to zero as the size of the training set is increased.

3.2.2. Gabor Patch Task

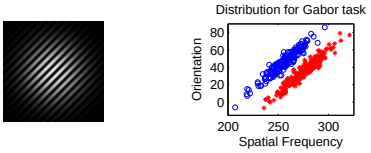
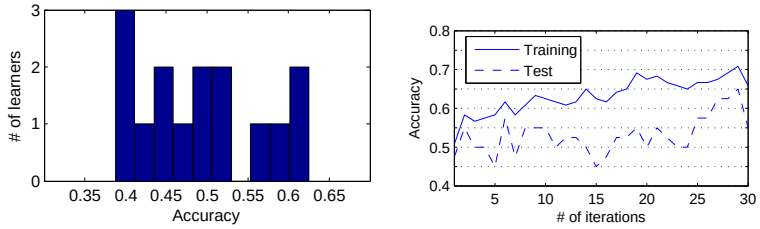


Figure 6. Gabor Learning Task

A Gabor patch(Figure 6(a)) is a sinusoidal grating with control parameters: spatial frequency d and orientation θ. The patches are generated using a sine wave $z = sin((0.25 + d/50)x)$ rotated θ anticlockwise from the positive x-axis and presented as a 200×200 pixel image. These dimensions are separable so humans can reason independently about each of these dimensions but they find it difficult to learn decision rules jointly involving both dimensions. This task is an example of learning information-integration boundaries, albeit around 800 examples are required to learn the information-integration boundary. The classes are drawn as in Figure 6(b) where the optimal decision boundary is linear at a 45° angle to the axes. We use parameter values as in (Ell et al., 2009), which were chosen such that the resulting patches are not too dense and it is possible to count the number of lines in the grating and the effects of the dimensions are balanced. Since humans naturally find this task difficult, we do not add noise. Note that an ideal boosting procedure using decision stumps on the true parameters can achieve 100% accuracy on this task. Thus, this task is one of learning an information-integration boundary using an aggregation of rule-based learners. We demonstrate that boosting excels at this task and achieves very high accuracy.

In preliminary experiments, we found that if the deci-



(a) Control group accuracy (b) Boosting accuracy

Figure 5. Boosting on the Opinion Spam dataset: Due to the complexity of the task, control group learners can identify relevant features, but not the class towards which they point. After 30 rounds, the boosted learner performs better than all control group learners, though the accuracy will increase if we run it for more iterations

sion boundary is axis-parallel, then it is learned perfectly with < 10 examples. On the distribution shown in Figure 6(b), the average accuracy of human learners from the control group trained with 20 examples each was 60.11% $\pm$ 12.22%(see Figure 3(a)). Most learners learned decision rules involving the spatial frequency while none used the orientation. Some participants learned rules based on irrelevant artifacts such as "ropiness" of lines and their responses could not discriminate between classes. The optimal spatial frequency decision boundary achieves accuracy of 72%, achieved by 3 learners, though 100% performance is possible in principle.

Boosting with 20 training examples(Figure 3(b)) quickly achieves 72% training accuracy and stabilizes at this value (first 11 rounds shown). This is a failure of the boosting algorithm as it only performs as well as the best human learner. Examining the user responses revealed that all learned decision rules used only the spatial frequency. We speculate that this is because, with 20 examples, learners focus on the most obvious discriminative perceptual dimension(spatial frequency) and ignore the others. Boosting with 5 training examples(Figure 3(b)) on the other hand allows weak learners to learn decision rules based on the orientation and consequently achieves $> 90\%$ training accuracy. This finding might be of interest even from a cognitive science point of view. It is interesting to note that (Friedman, 2002) observed a similar effect with decision trees, viz subsampling can improve performance, though their differences were not as stark.

3.2.3. Opinion Spam Task

While the earlier experiments were based on synthetic data, we now consider the performance of our method on a real world dataset. The Opinion Spam task aims to separate truthful and deceptive reviews. Each stimulus is a hotel review, usually a small paragraph with 4-5 sentences. We used the opinion spam detection dataset introduced in (Ott et al., 2011) which contains truthful hotel reviews extracted from TripAdvisor and deceptive reviews for the same hotels generated using Mechanical Turk. By design, deceptive reviews are meant to deceive humans and Turkers do find this task difficult, often performing near random. This task is challenging because of the number of possible perceptual dimensions to the task, because the nature of the decision boundary is unknown and because the classes may overlap. This task is particularly interesting for a black-box procedure such as boosting, since the relevant features are unknown. The decision boundaries learned by human learners are based on semantic features which are difficult to capture with computers (is the review humorous?, sarcastic?, well-written?). Preliminary experiments showed that if the learners are informed that they are trying to separate fake and truthful reviews, they completely ignore the given training set and use their own preconceived notions of what constitutes a truthful review. Hence, in our experiments they are only informed that the task is to separate two classes of reviews. This may, however, lead to many features not being discriminative. The average accuracy over the control group human learners (Figure 5(a)) trained with 10 examples was 51% $\pm$ 8.76%, i.e. only slightly better than random. In this task, we only expect human learners to identify discriminating features in the training set and to classify the remaining instances according to that feature.

With human boosting(Figure 5(b)), we gradually achieve 70% training accuracy and 65% test accuracy on this task after 30 boosting rounds. We expect the increasing trend in the training accuracy to continue if the algorithm is run for more iterations.

Acknowledgments

We acknowledge the support of NSF via IIS-1149803, and DoD via W911NF-12-1-0390.

References

Amazon mechanical turk. `http://www.mturk.com`.

Ashby, F.G. and Maddox, W.T. Human category learning. *Annu. Rev. Psychol.*, 56:149–178, 2005.

Ashby, F.G. and Maddox, W.T. Human category learning 2.0. *Annals of the New York Academy of Sciences*, 1224(1):147–161, 2011.

Ashby, F.G., Maddox, W.T., and Bohil, C.J. Observational versus feedback training in rule-based and information-integration category learning. *Memory & cognition*, 30(5):666–677, 2002.

Ben-David, S., Long, P., and Mansour, Y. Agnostic boosting. In *CoLT*, pp. 507–516. Springer, 2001.

Bradley, J. K. and Schapire, R. Filterboost: Regression and classification on large datasets. *NIPS*, 20: 185–192, 2008.

Collins, M., Schapire, R.E., and Singer, Y. Logistic regression, adaboost and bregman distances. *Machine Learning*, 48(1):253–285, 2002.

Cosmides, Leda and Tooby, John. Are humans good intuitive statisticians after all? rethinking some conclusions from the literature on judgment under uncertainty, 1996.

Dawid, A.P. and Skene, A.M. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, pp. 20–28, 1979.

Dietterich, T. G. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157, 2000.

Domingo, C. and Watanabe, O. Madaboost: A modification of adaboost. Citeseer, 2000.

Ell, S., Ing, A., and Maddox, W. Critrial noise effects on rule-based category learning: The impact of delayed feedback. *Attention, Perception, & Psychophysics*, 71:1263–1275, 2009. ISSN 1943-3921.

Fried, L.S. and Holyoak, K.J. Induction of category distributions: A framework for classification learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 10(2):234, 1984.

Friedman, J. H. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

Friedman, J. H., Hastie, T., and Tibshirani, R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *Ann. Stats.*, 28(2):337–407, 2000.

Grubb, A. and Bagnell, J.A. Generalized boosting algorithms for convex optimization. *arXiv preprint arXiv:1105.2054*, 2011.

Hsu, A.S., Griffiths, T.L., et al. Effects of generative and discriminative learning on use of category variability. 2010.

Kearns, M. and Valiant, L. Cryptographic limitations on learning boolean formulae and finite automata. *JACM*, 41(1):67–95, 1989.

Mason, L., Baxter, J., Bartlett, P.L., and Frean, M. Functional gradient techniques for combining hypotheses. *NIPS*, pp. 221–246, 1999.

Meir, R. and Rätsch, G. An introduction to boosting and leveraging. *Advanced lectures on machine learning*, pp. 118–183, 2003.

Ott, M., Choi, Y., Cardie, C., and Hancock, J.T. Finding deceptive opinion spam by any stretch of the imagination. *Arxiv preprint arXiv:1107.4557*, 2011.

Quinn, A.J. and Bederson, B.B. Human computation: a survey and taxonomy of a growing field. In *CHI*, pp. 1403–1412. ACM, 2011.

Schapire, R.E. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

Schapire, R.E. The boosting approach to machine learning: An overview. *Lecture Notes in Statistics*, pp. 149–172, 2003.

Settles, B. Active learning literature survey. *University of Wisconsin, Madison*, 2010.

Sheng, V.S., Provost, F., and Ipeirotis, P.G. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 614–622. ACM, 2008.

Zadrozny, B., Langford, J., and Abe, N. Cost-sensitive learning by cost-proportionate example weighting. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pp. 435–442. IEEE, 2003.

Zhu, X., Gibson, B.R., Jun, K.S., Rogers, T.T., Harrison, J., and Kalish, C. Cognitive models of test-item effects in human category learning. In *ICML*, 2010.

Zhu, X., Gibson, B.R., and Rogers, T.T. Co-training as a human collaboration policy. In *AAAI*, 2011.