Appears in Proceedings of AAAI 2013

# Teaching Classification Boundaries to Humans

**Sumit Basu**
Microsoft Research
Redmond, WA
sumitb@microsoft.com

**Janara Christensen**
University of Washington
Seattle, WA
janara@cs.washington.edu

## Abstract

Given a classification task, what is the best way to teach the resulting boundary to a human? While machine learning techniques can provide excellent methods for finding the boundary, including the selection of examples in an online setting, they tell us little about how we would teach a human the same task. We propose to investigate the problem of example selection and presentation in the context of teaching humans, and explore a variety of mechanisms in the interests of finding what may work best. In particular, we begin with the baseline of random presentation and then examine combinations of several mechanisms: the indication of an example's relative difficulty, the use of the shaping heuristic from the cognitive science literature (moving from easier examples to harder ones), and a novel kernel-based "coverage model" of the subject's mastery of the task. From our experiments on 54 human subjects learning and performing a pair of synthetic classification tasks via our teaching system, we found that we can achieve the greatest gains with a combination of shaping and the coverage model.

## Introduction

Machine learning has yielded a broad variety of impressive results on how best to train a classifier, but what is the best way to teach a classification task to a human? The literature has mostly neglected this question, despite the potential for education as well as other scenarios. This raises the question of why one might wish to teach humans this particular type of task. Beyond the underlying questions about how best to teach human subjects in this scenario, we believe there are several practical applications. The first and most direct is that of teaching a real-world discrimination task. As humans we are often required to make such classifications: is this a safe link to click on or not, is this a safe place to use a credit card, is this peach sufficiently ripe, is this a morel mushroom or a poisonous false morel – if we can do better than picking random examples for teaching such tasks, we may be able to improve human performance in the real world. A second scenario that is increasingly important is training (calibrating) human labelers for high inter-annotator agreement with past labelers. By teaching new raters using labels from past raters, we can train them to produce more consistent and thus higher quality data.

We know from both theoretical and empirical findings that classifiers learn best from data that constitute a representative sample, i.e., a random sample that has the same distribution as the target data. There are variations such as active learning, in which a classifier can pick from a palette of unlabeled points to be labeled. Humans are quite different from classifiers, though: simply seeing representative examples does not mean they will perform optimally with respect to them. They will be more sensitive to some features than others, they will not perfectly consider all the information from all the samples, and they will be confused by items that are similar. They may have "blind spots" or inherent biases that lead them to make less than optimal decisions. Active learning approaches where the human could pick the next sample to be labeled may help in limited scenarios, but when there are multiple dimensions (of features), the perplexity of possible choices could prove overwhelming. Similarly, automatically choosing examples that would be "best" for a classifier can be disastrous for a human; we found this in our own pilot experiments.

Given the possible benefits and inherent questions, we investigate in this paper what might be the best way to teach a classification task to a human from a palette of available methods. The obvious baseline is random presentation, i.e., picking examples from the same distribution as the test set, but we suspected that it would be possible to do better. We thus developed a set of mechanisms we hypothesized could be helpful to humans in the learning task, some based on the cognitive science and machine learning literature, others based on our models of task mastery. In particular, these mechanisms were (1) an indicator of individual example difficulty, (2) a means of selecting examples in order of increasing difficulty (curriculum learning), and (3) the estimation of a "coverage model" of the subject's mastery over the data space and then sampling from its complement. To test the effectiveness of our methods,

we developed a synthetic learning task: subjects learned to classify parametrically drawn mushrooms based on presented examples and then took a quiz to determine how much they learned. As a result of these experiments, we found that indeed we can do substantially better than random presentation with some combinations of these methods.

## Related Work

There is a wealth of literature on the general area of teaching with the help of computers, an area known as Interactive Tutoring Systems, but most of this work is focused on aiding with traditional classroom curricula; the recent compilation of Woolf (2008) provides an excellent overview. The closest to our topic is the work on teaching categories, specifically in the context of teaching vocabulary. The emphasis in that research has been on spacing effects, i.e., selecting the interval of study to maximize retention; see for instance the recent work of Mozer et al. (2009). If we reach further afield into the cognitive science literature, there is much work examining how humans form and learn categories, like the classic work of Markman (1989).

The closest research to our own appears in the machine learning literature, in an investigation by Castro et al. (2008) which sought to examine how active learning heuristics would fare in the context of teaching humans. As active learning is focused on finding the best order of examples to present to an online learning algorithm (see (Settles 2010) for a current review of active learning), it was a natural choice to apply these algorithms to humans. The authors presented subjects with a 1-D (single parameter) classification problem, where their goal was to identify the location of the boundary. Subjects were shown examples with the parameter illustrated with synthetic "eggs" of varying shapes. They found the best results came via an approach they termed "yoked learning," in which they bisected the posterior for the location of the boundary at each step given the presented examples, getting ever closer to the solution. We note that in 1-D, the boundary is just a point, whereas in higher dimensions (such as in our experiments), the boundary is an N-1 dimensional hyperplane, and there is no corresponding bisection approach. While such a method could be used to find a single point on the boundary, our goal is to teach the human the difference between the classes, and for that they need to know the entire classification boundary.

Another approach Castro et al. found helpful was letting humans choose the next location to view. Again, that makes sense in a 1-D space – subjects can quickly converge on the boundary. In the 4-D space of our experiments there are far too many possibilities, and no simple way for subjects to visualize the space. In our study, we had 10,000 points sampled uniformly from the space, and showing all of these or even a representative sample in a random ordering would be quite overwhelming. Furthermore, we were also interested in the case where the feature space could be latent, i.e., the correspondence to a vector space is not explicit, as with a real-world classification task with only a similarity metric available, and as such, we didn't want to focus on methods that took advantage of an explicit representation of the feature space. We also note that their methods were independent of the user's performance during training. In their workflow, the user was shown labels for the examples they or the algorithm picked (users were tested at intervals on the location of the boundary); as such their performance was not recorded or used. In our work, with every training example the subject was required to test their knowledge before the label was revealed. Our work leverages this information during training via our "coverage model."

Another thread of related work comes from the "shaping" work in the classic cognitive science literature via its recent application in the machine learning community. The original work, by Skinner and others (Peterson 2004) was based on the theory that it is possible to teach a complex task/behavior by building up and reinforcing smaller subtasks. A generalization of this theory is that it may be more efficient to teach a difficult concept by starting off with easy versions of it (Kreuger and Dayan 2009). The latter interpretation has found success in recent work on online learning, in which easier examples are presented before more difficult examples for online training of classifiers; this approach is known as "curriculum learning," as explored in (Bengio et al. 2009) and (Kumar, Packer, and Koller 2010). Given the success of this approach in teaching classifiers (recently) and animals/humans (classically), we added this strategy for teaching to our suite. The work on curriculum learning led Khan, Zhu, and Mutlu (2011) to a related investigation, though with the converse of our goals, in which the authors studied how humans might choose to teach a classifier (robot).

In summary, other than (Castro et al. 1980), ours is the only work we are aware of to teach classification boundaries, and the first to teach higher dimensional boundaries.

## Learning Task

For this study, we wanted to choose a classification task of sufficient complexity that users would not master it with a few examples, but not so difficult that it would take egregiously long to learn. We also wanted to be able to generate a large number of unique learning problems of equivalent difficulty; as such we focused on using synthetic data with randomly selected boundaries. Through extensive piloting, we found 4-dimensional data to be a good compromise: subjects could quickly get a good understanding of the problem without saturating their performance too early.

As we wished to make the task as natural as possible, we chose to represent the data as parametrically drawn mushrooms instead of directly showing the numerical values of the four-dimensional instances (see Figure 1). The four di-

mensions, all between 0 and 1, were mapped to ranges of the stem width, stem height, cap width, and the cap height.

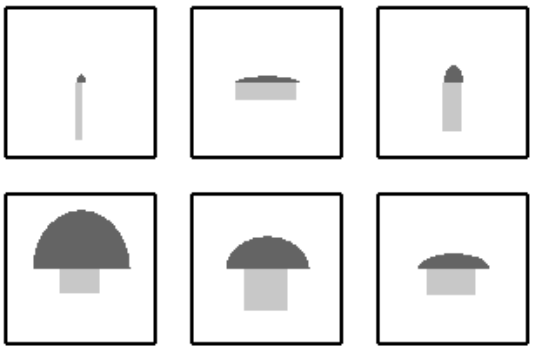

**Figure 1: Examples of parametrically drawn mushrooms.**

The classification boundary was an N-1 dimensional hyperplane; we did not add any label noise, and as such the problems were linearly separable. In order to ensure that the tasks would be of equal difficulty and that random guessing would result in a base rate of 50% on average, we required the boundary to pass through the origin. Formally, we randomly chose a 4-dimensional $w$, normalized it to unit length, and computed the label $y$ for each exemplar $x$ as 1 if $w^T x \geq 0$ and 0 otherwise. As such, we were able to construct a new problem for every task and every subject: for each trial we would begin by sampling 10,000 points from a uniform distribution and then generating a boundary.

## Teaching Methods

As our goal was to investigate what methods might work best for teaching, we developed several mechanisms to compete with our baseline of random presentation.

### Baseline

We initially considered two baseline methods, random presentation and the active learning heuristic of the points closest to the boundary. We initially included the boundary method as well because it seemed like a good instance of a scheme that was ideal for a learning algorithm but terrible for human learners attempting to perform a classification task. Our intuition was correct: in an experiment with a three-dimensional feature space (also represented as a mushroom), a set of 13 users achieved 53.6% performance, far worse than random presentation (on which they achieved 71.7%), a difference that was significant (two-tailed t-test, unequal variances: p=0.001, df=22.7, t=3.76). Furthermore, subjects complained bitterly about how frustrating and confusing this condition was. As its inferiority was clear, we left out this method for the final experiment.

### Difficulty Indicator

The first mechanism we considered was an indicator for the difficulty of a given training example. This was shown as a small colored square in the upper right corner of the view (see Figure 2). The color varied from bright green (easy) to bright red (hard). The difficulty value was computed as the negative of the projected absolute distance to the boundary, normalized to be between 0 (easiest) and 1 (hardest):

$$d = \frac{z - |w^T x|}{z}$$

In this formulation, $z$ is a normalizing constant equal to the maximum possible value for $w^T x$ given the choice of $w$.

### Curriculum Learning

Our next proposal originated from the cognitive science work in shaping as well as the machine learning work in curriculum learning, i.e., the idea that one could train a learner more efficiently by presenting easier examples first and then showing progressively harder examples. For our study, we used the same measure shown above for the difficulty indicator. For any given training example number, we chose a $d_{max}$ linearly related to the example number, such that the maximum hardness was achievable by the 20th example (out of 30). Examples were still drawn randomly, but if the drawn instance had difficulty greater than $d_{max}$, it was returned to the set and a new example was chosen. This process continued until an example was found with difficulty less than $d_{max}$. We also added a balance constraint that prevented the difference between the number of seen positive and negative examples to be no more than one.

### Coverage Model

Our final scheme for improving learning involved what we term a "coverage model:" we model the areas of the example space in which the subject had gained expertise, and then sample from the complement of that distribution. This method has analogies to the boosting approach of classifier combination, in which examples where the classifier performs poorly are reweighted to focus the efforts of successive learners (Freund and Schapire 1999). As we are not doing batch learning, we cannot explicitly apply the reweighting used in boosting, but instead use a kernel-based approach to represent the expertise distribution and use it to guide example selection.

Since points near each other would be similar in mushroom appearance and typically in label, we modeled the user's knowledge as a mixture of Gaussian kernels of fixed variance. The kernel centers were at every point the subject had been given a training example, as well as 100 random samples chosen from the unlabeled points to prevent overfitting. The form of the function was as follows:

$$p_c(x) = \frac{1}{z}\sum_{i \in l,r} \alpha_i \mathcal{N}(x; x_i, \sigma^2) \qquad \alpha_i \geq 0$$

The set $l$ represents the indices of the examples that have been labeled by the subject during training, the set $r$ is the

randomly selected set of 100 unseen examples, $z$ is a normalizing constant such that the distribution sums to one over all possible examples, and $\sigma^2$ is a fixed kernel width that we determined via pilot experiments. We require the $\alpha_i$ to be positive to ensure that $p_c$ is positive everywhere.

We define the probability of coverage (user knowledge) to be 1 on those indices in $l$ that the user labeled correctly, 0 on those that she labeled incorrectly, and 0.5 on those in $r$ (which have not been seen yet). Since we only have target values at these points, we can solve this as an $l + r$ dimensional discrete minimization problem:

$$L(\alpha) = \|b - A\alpha\|^2 \qquad A[i,j] = \mathcal{N}(x_i;\, x_j, \sigma^2)$$

In this expression, $L(\alpha)$ is the loss function, and $b$ is the vector of target probabilities. To constrain the $\alpha_i$ to be positive, we parametrize them as $\gamma_i^2$. We can then take partial derivatives of the loss with respect to $\gamma$ via the chain rule:

$$\frac{dL}{d\alpha} = 2A^T(A\alpha - b) \qquad \frac{d\alpha_i}{d\gamma_i} = 2\gamma_i$$

Given these partials, we use L-BFGS Quasi-Newton optimization (Nocedal 1980) to minimize the loss.

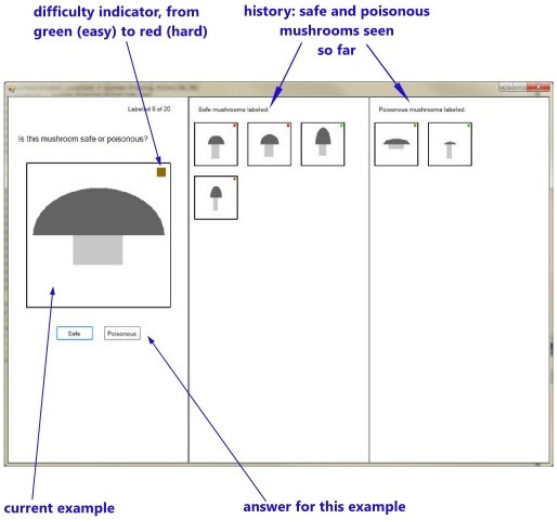


**Figure 2: The teaching interface.**

## Teaching Interface

We designed a teaching interface to test the different methods with our human subjects (Figure 2), consisting of three vertical panes. During training, the leftmost pane shows the current training example, along with buttons with which the subject can indicate whether it is "safe" or "poisonous." The interface then reveals the answer, and places the example into the appropriate (safe/poisonous) history pane. The subject can thus review past training examples when making future judgments during both training and testing.

There are some important differences to the workflow during testing. First, the subject is not told whether her individual answers are correct, nor are examples added to the history panes; this is to prevent the subject from doing further training during the testing period. Finally, the hardness indicator is never shown on new examples during testing.

## Experiments

We recruited a total of 90 subjects at our institution; 54 completed the entire study. As we found individuals' capabilities varied widely in our pilot studies, we used a *within-subjects* design in which each subject went through two train/test conditions, the baseline and one other condition. Each condition had a different class boundary for each user. In this way, we were able to measure the relative performance of a given individual between the baseline and one of our methods. To protect against ordering effects, we presented the two conditions in random order.

There were a total of five teaching methods; each subject experienced the baseline as well as one other method.

- *Base*: training examples were chosen randomly.
- *Ind*: the difficulty indicator was shown; examples were chosen randomly.
- *Covg:* the coverage model was used to select examples.
- *IndCurr*: the difficulty indicator was shown, and examples were filtered by difficulty using the curriculum learning approach as described above.
- *IndCurrCovg*: the difficulty indicator was shown; examples were chosen via the coverage model and also filtered by difficulty via curriculum learning.

Subjects were divided into the four possible condition pairs based on the order they started the task; as not all completed the task, there is some variance in the number of subjects (see Table 1). Each subject completed two learning tasks, each with 30 training examples followed by 20 test; the test examples were always drawn from a uniform distribution. They then filled out a short survey about the tasks.

## Results and Analysis

Given the data from the subjects, we sought to understand the differences in performance as well as what may have led to those differences. Below, we consider subjects' performance and perceptions under the various conditions.

### Relative Performance by Condition

The core results of our study are in terms of relative performance on the test set between each of the teaching conditions and the baseline. We used the dependent t-test for paired samples (two-sided) to compare the subjects' performance in the two conditions they saw (Table 1). The first two (*Ind* and *Covg*) did not show significant differences from the baseline, though *Ind* had a nearly-significant negative effect. While not conclusive, it seems possible that the difficulty indicator on its own may have hurt subjects' performance. We hypothesize this might have been due to subjects ignoring more difficult training examples as being too hard, or being frustrated by wildly fluctuating difficulty.

| Method | p | d.f. (N-1) | t | Improvement |
|---|---|---|---|---|
| Base vs. Ind | 0.12 | 12 | -1.71 | -0.09 |
| Base vs. Covg | 0.43 | 13 | -.81 | -0.05 |
| Base vs. IndCurr | **0.08** | **15** | **1.88** | **0.08** |
| Base v. IndCurrCovg | **0.007** | **10** | **3.36** | **0.12** |

***Table 1*: Test performance improvements (mean differences in per-condition accuracy scores across subjects) with respect to baseline. Statistically significant results (p<0.1) are in bold.**

The last two conditions, *IndCurr* and *IndCurrCovg*, did produce significant improvements; the strongest effect was for *IndCurrCovg*, both in magnitude (12 points absolute improvement, from 61% to 73%) and in significance (p=0.007). This implies that the coverage model in conjunction with curriculum learning has a stronger effect on performance than either on their own. In the remainder of our analysis, we will examine the fine-grained differences in behavior and performance to better understand these results.

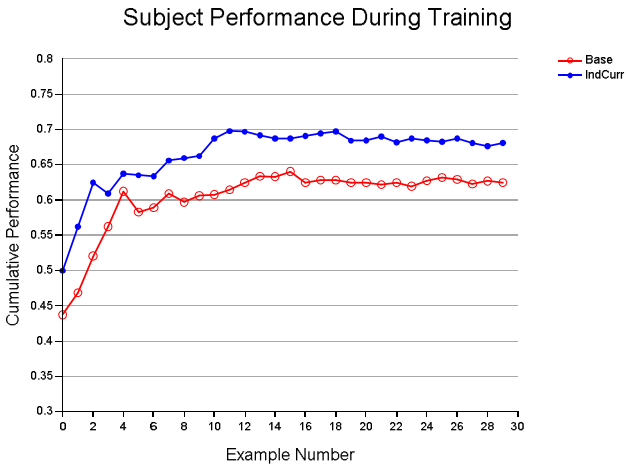


**Figure 3: Mean cumulative subject performance during training for the *Base* vs. *IndCurr* condition (N=16).**

## Performance During Learning

Our first area of investigation involves seeing how subjects performed during training by looking at their cumulative training performance (over all examples presented so far). In Figures 3 and 4, we look at the mean performance across subjects for the *IndCurrCovg* and *IndCurr* conditions.

In both sets of conditions, we see that when subjects are in the baseline condition, their performance improves for the first few examples, but then saturates by example 15 or so. For *IndCurr*, they do better earlier on and improve in the same way but also saturate in performance at a similar point, albeit at a higher level. For *IndCurrCovg*, on the other hand, we see what appears to be a faster and longer rise in performance before learning saturates.

While both the *IndCurr* and *IndCurrCovg* conditions have the benefit of curriculum learning, we believe the greater gain in the latter is coming from the coverage model's ability to model and highlight areas the subject performed poorly in or had not covered. This may be allowing the subject to continue learning longer, whereas the other models may start presenting examples that are redundant.

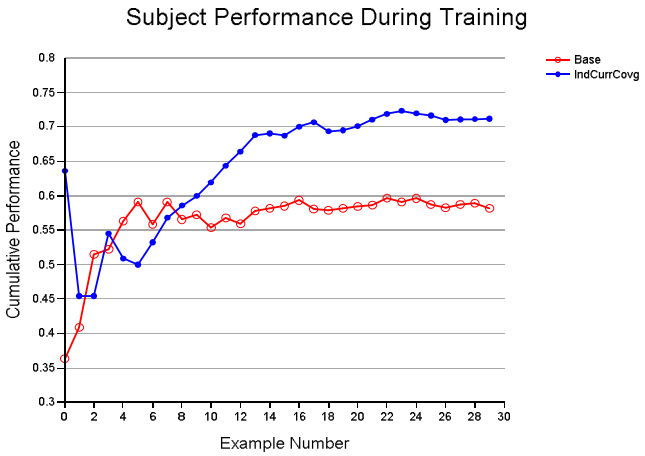


**Figure 4: Mean cumulative subject performance during training for the *Base* vs. *IndCurrCovg* condition (N=11).**

## Distribution of Training Example Difficulty

Given the difference in effect of the coverage model in the *IndCurrCovg* condition vs. the *Covg* condition, we suspected that without restriction, the model might quickly converge onto the most difficult examples and confuse the user. In Figure 5, we examine the distribution of examples for *Base*, *Covg*, and *IndCurrCovg* over quartiles of difficulty. For *IndCurrCovg*, it is biased towards easier examples. For *Covg*, though, it is indeed biased in the *other* direction.

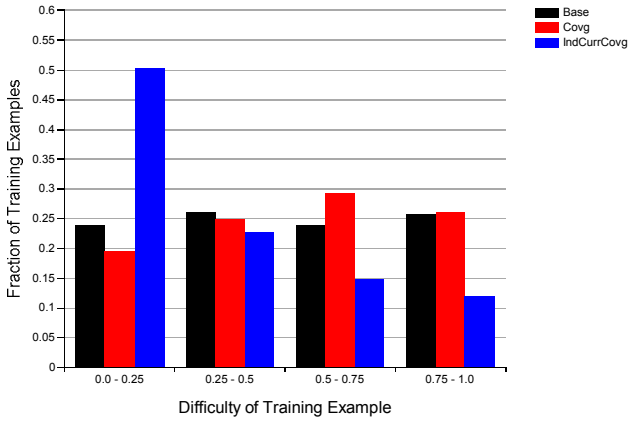


**Figure 5: Distributions of training example difficulties.**

## Performance vs. Example Difficulty

We next wished to investigate how the different strategies affected performance at different levels of example difficulty during testing. We plotted the relative performance for

condition pairs over quartiles of difficulty in Figures 6-8. Both *IndCurrCovg* and *IndCurr* show improvements over *Base* across all levels. We also show the relative performance for the *Base* vs. *Covg* condition in Figure 8 to investigate why the coverage model was not helpful on its own. Note that while performance for medium hardness increased slightly, performance on the easiest examples dropped substantially. This is potentially a consequence of the more difficult training examples shown in this case (see Figure 5); the fact that the users did *worse* on the *easiest* examples implies they did not learn the concept well at all.

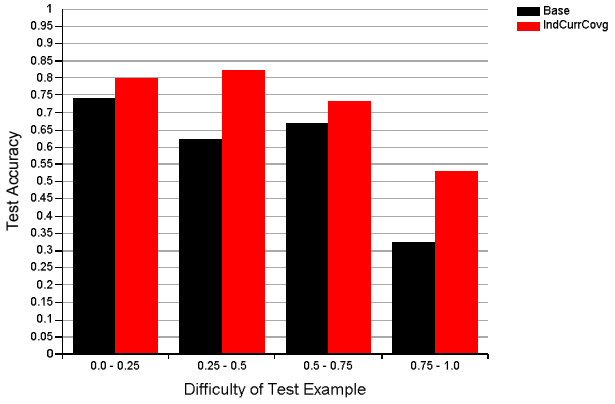


**Figure 6: Test perf. vs. difficulty: *Base* vs. *IndCurrCovg*.**

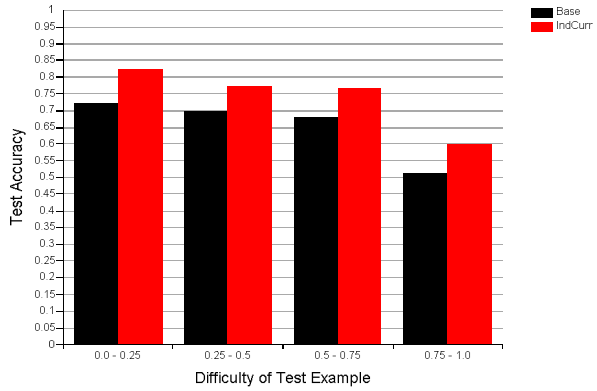


**Figure 7: Test performance vs. difficulty: *Base* vs. *IndCurr*.**

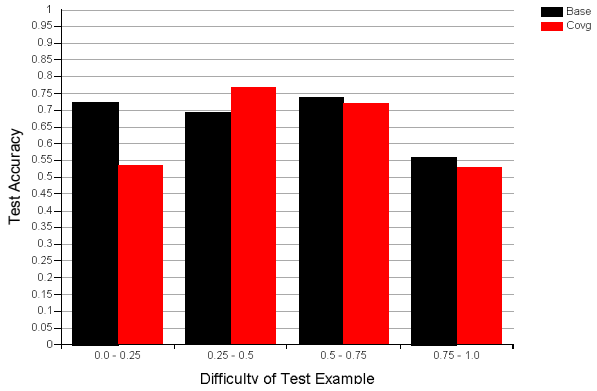


**Figure 8: Test performance vs. difficulty: *Base* vs. *Covg*.**

## Survey Results

Finally, we examine the qualitative feedback from users. In Table 2 we look at the answers to "which condition was more enjoyable?" and in Table 3 "which condition was easier?" Subjects overwhelmingly felt the *IndCurrCovg* condition was better in both respects: the mechanism that most improved performance, then, also improved most user experience, perhaps because of the greater feeling of mastery. We see this to a lesser extent for *IndCurr* as well.

The results for the other two methods were also interesting: subjects were split on both questions for *Base* vs. *Covg*, consistent with their performance. In *Base* vs. *Ind*, on the other hand, subjects overwhelmingly preferred the baseline. This adds to our earlier suspicions that the difficulty indicator may have led to confusion and/or frustration.

| | **Base** | **Other** | **Same** |
|---|---|---|---|
| Base vs. Ind | **9** | 3 | 0 |
| Base vs. Covg | 5 | **7** | 2 |
| Base vs. IndCurr | 2 | **12** | 3 |
| Base vs. IndCurrCovg | 3 | **8** | 0 |

***Table 2*: Survey results for the question "Which condition was more enjoyable?**

| | **Base** | **Other** | **Same** |
|---|---|---|---|
| Base vs. Ind | **9** | 3 | 0 |
| Base vs. Covg | 5 | **6** | 3 |
| Base vs. IndCurr | 4 | **11** | 2 |
| Base vs. IndCurrCovg | 1 | **9** | 1 |

***Table 3*: Survey results for the question "Which condition was easier?"**

## Discussion and Future Work

While teaching classification tasks to humans is different from teaching algorithms, it is encouraging to see that some ideas from machine learning are indeed helpful. Given the strong, highly significant ($p=0.007$) effect on performance, we can say with some confidence that the combination of the difficulty indicator, curriculum learning, and the coverage model (*IndCurrCovg* strategy) does improve subjects' performance at learning classification boundaries. With less confidence, we can say that the *IndCurr* strategy (difficulty indicator and curriculum learning) improves performance as well. As such, it seems the combination of curriculum learning and the coverage model are more powerful than either on its own, as the *Covg* condition did not produce a significant change in performance. As we had suspected upon seeing the results, the coverage model on its own seemed to emphasize examples that were too hard, whereas when coupled with curriculum learning it helped subjects

address their areas of difficulty while not overwhelming them with hard examples early on.

The difficulty indicator, while intuitively a helpful additional source of information about each training example, seemed to both reduce subject's performance (though not quite statistically significant in effect) and their enjoyment, as well as increasing their perception of task difficulty. We hypothesize that subjects became frustrated with and/or ignored examples marked as difficult, and suspect this condition was particularly vexing in the absence of curriculum learning, since hard and easy examples would be appearing in random sequence. Perhaps seeing the hardness increase in a controlled way (in the *IndCurrCovg* condition) reduced or removed the negative effects of the indicator. However, it's also possible that removing the indicator and running a *CurrCovg* condition would show even greater gains; we hope to investigate this in future work. It may be that subjects are happier and learn better when they simply don't know the relative difficulty of the training example; this could have interesting ramifications for other educational domains as well.

As we outlined earlier, we believe there are a variety of applications for which teaching classification boundaries to humans can be important: we hope these methods and results can be helpful in such contexts, but also that they may extend to a broader set of teaching problems.

## References


Andrews, R., Diederich, J., and Tickle, A.B. Survey and Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks. *Knowledge Based Systems*. 8(6) 373-389. (1995).

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum Learning. In *Proceedings of ICML* 2009 (2009).

Castro, R., Kalish, C., Nowak, R., Qian, R., Rogers, T., and Zhu, X. Human Active Learning. In *Proceedings of NIPS 2008* (2008).

Freund, Y. and Schapire, R. E. A Short Introduction to Boosting. *Journal of the Japanese Society for Artificial Intelligence* 14, 5 (1999), 771-780.

Khan, F., Zhu, X., and Mutlu, B. How Do Humans Teach: On Curriculum Learning and Teaching Dimension. In *Proceedings of NIPS 2011* (2011).

Kreuger, K.A., and Dayan, P. Flexible Shaping: How Learning in Small Steps Helps. *Cognition* 110, 3 (2009), 380-94.

Kumar, M.P., Packer, B., and Koller, D. Self-Paced Learning for Latent Variable Models. In *Proceedings of NIPS 2010* (2010).

Markman, E. M. *Categorization and Naming in Children: Problems of Induction*. MIT Press, (1989).

Mozer, M.C., Pashler, H., Cepeda, N., Lindsey, R., and Vul, E.. Predicting the optimal spacing of study: A multiscale context model of memory. In *Proceedings of NIPS 2009* (2009).

Nocedal, J. Updating Quasi-Newton Matrices with Limited Storage. Mathematics of Comp. 35 (1980), 773-782.

Peterson, G.B. A Day of Great Illumination: B.F. Skinner's Discovery of Shaping. *Journal of Experimental Analysis of Behavior* 82 (2004), 317-28.

Settles, B. Active Learning Literature Survey. *University of Wisconsin-Madison Computer Science Technical Report 1648*, (2010).

Woolf, B.P. *Building Intelligent Interactive Tutors.* Morgan Kaufmann. (2008).