arXiv:1802.07427v1 [cs.LG] 21 Feb 2018

Active Learning with Partial Feedback

Peiyun Hu[1]*, Zachary C. Lipton[1,3], Anima Anandkumar[2,3], Deva Ramanan[1]

Carnegie Mellon University
California Institute of Technology
Amazon AI

peiyunh@cs.cmu.edu, zlipton@cmu.edu, anima@caltech.edu, deva@cs.cmu.edu

February 21, 2018

Abstract

In the large-scale multiclass setting, assigning labels often consists of answering multiple questions to drill down through a hierarchy of classes. Here, the labor required per annotation scales with the number of questions asked. We propose active learning with partial feedback. In this setup, the learner asks the annotator if a chosen example belongs to a (possibly composite) chosen class. The answer eliminates some classes, leaving the agent with a *partial label*. Success requires (i) a sampling strategy to choose (example, class) pairs, and (ii) learning from partial labels. Experiments on the *TinyImageNet* dataset demonstrate that our most effective method achieves a 21% relative improvement in accuracy for a 200k binary question budget. Experiments on the *TinyImageNet* dataset demonstrate that our most effective method achieves a 26% relative improvement (8.1% absolute) in top1 classification accuracy for a 250k (or 30%) binary question budget, compared to a naive baseline. Our work may also impact traditional data annotation. For example, our best method fully annotates TinyImageNet with only 482k (with EDC though, ERC is 491) binary questions (vs 827k for naive method).

1 Introduction

Given a large set of unlabeled images, and a budget to collect annotations, how can we learn an accurate image classifier most economically? Given access to examples $\boldsymbol{x}$ and ignoring computation, costs depend upon the annotation procedure. Active Learning (AL) offers a route to increase efficiency by strategically choosing which examples to annotate, AL gains by sampling more *informative* examples.

Classical AL formulations treat the labeling process as atomic: every query produces an *exact label* and comes at equal cost. However, as Settles (2011) notes, for real-world problems, costs often vary across examples. Labels are commonly related via a hierarchy of concepts, especially when there are a large number of classes Lipscomb (2000); Miller (1995). In these cases, practitioners collect exact labels by asking multiple questions to drill down through the hierarchy. Consider an image dataset containing atomic labels like *golden retriever*, *bulldog*, *Persian cat*, and *screwdriver*. One high-level question might be, “Does

*This work was done while the author was an intern at Amazon AI

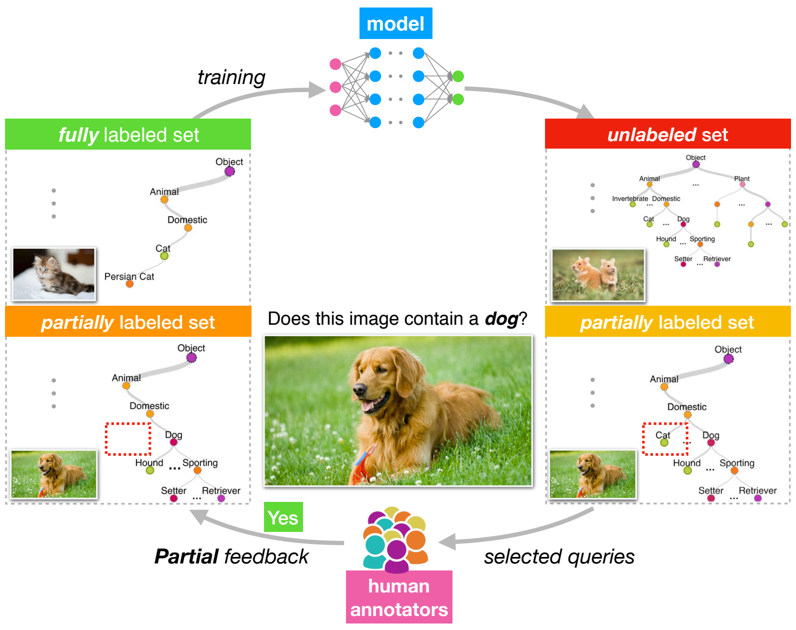


Figure 1: Active learning with partial feedback.

the image depict a dog?" The answer *yes*, provides *partial feedback*, ruling out *screwdriver* and *Persian cat*. We denote the resulting subset of feasible labels as a *partial label*.

When labels are solicited by drilling through a hierarchy, the true cost may be the number of questions required to produce a class label and that cost might scale proportionate to the number of questions asked. In the naive drill-down strategy, images belonging to classes with a shorter distance from root to leaf may be cheaper to annotate.

In this paper, we propose the problem of *active learning with partial feedback* (ALPF). Here, the learner must choose both *which examples* to annotate, and *which questions* to ask. ALPF differs from classic active learning in the following ways: (i) at each round, the ALPF learner selects an (example, class) pair; (ii) The annotator provides binary feedback, leaving the learner with a *partial label* and when only the true atomic class remains, the learner has obtained an *exact label*; (iii) In addition to the atomic classes of interest, the learner possesses a pre-defined collection of *composite classes* (e.g. in a hierarchy *dog sup bulldog*, *mastiff*, ...); and (iv) the learner updates its model given both partial and exact labels. Efficient learning requires both an algorithm for learning from partial labels, and choosing a superior sampling strategy.

Binary feedback mechanisms are appealing because (i) they are already in wide use and (ii) they can collect answers quickly, requiring only one button (click: 1, no-click: 0). In contrast asking a user to select among 1000 classes might require that they spend considerable time scanning through the menu (perhaps a drop-down) of allowed classes. If the annotators do not have the set of class labels memorized, this might prove time consuming.

As a concrete example, consider an image for which our classifier confidently predicts the composite class *dog*, but remains uncertain as to the breed. The naive annotation scheme would start with top-level object categories, e.g. *natural* vs *artificial*, but an ALPF learner could skip this question and proceed directly to the dog breed.

We make the following experimental decisions: First, we focus on composite classes derived from an existing hierarchy. The justification for using only *real* composite classes is simple: while there are 2^k possible composite classes (assuming k atomic classes), real annotators cannot efficiently recognize *made-up* categories (composed of random atomic clases).

Objectives: Primarily we wish to optimize the traditional AL objective: given unlabeled examples and a labeling mechanism, **minimize error on the true distribution subject to a labeling *budget***. An extra benefit of ALPF is that it can help us to fully annotate a dataset quickly, cutting costs considerably by soliciting *exact labels* for all examples with fewer questions.

Methods: First we demonstrate an effective method for training deep neural networks (DNNs) given partial labels. In short: for each example, we convert the multiclass problem to a binary classification, where the two classes correspond to the subsets of *potential* (possibly applicable) and *eliminated* classes. We convert the softmax outputs over k classes into the estimated probability for a binary classification problem by marginalizing over the probabilities assigned to the *potential* class labels. The learning objective is then to maximize the log probability of the potential labels. The approach resembles previous schemes for learning from partial labels (Grandvalet & Bengio, 2004; Kakade et al., 2008), but is to our knowledge the first realization with either deep learning or active learning.

Our next methodological contribution is to propose several strategies for soliciting partial feedback: The first, *Expected Information Gain* (EIG), chooses those questions for which the entropy of the predicted distribution is expected to decrease most (upon receiving answer). We calculate the expectation using the current predictions. If the queried node is *dog* and the binary feedback says *yes*, then we set all non-dog classes to 0, and normalize the probabilities of classes belonging to the *dog* class. This heuristic generalizes the common AL heuristic of selecting examples with maximally entropic predictive distributions.

We introduce two sampling strategies, based on the expected change in the size of the *partial label*. The *Expected Decrease in Classes* (EDC) heuristic suggests choosing those questions that in expectation (based on current probability estimates) will eliminate the most atomic classes. The Expected Remaining Classes (ERC) sampling heuristic chooses questions for which the expected number of remaining *potential* classes following the question's answer is smallest. This method captures the hypothesis that we want as many exact labels as possible.

To validate our problem statement and to evaluate our proposed ALPF algorithms, we conduct extensive experiments using the CIFAR10, CIFAR100, and Tiny ImageNet datasets. In all cases, we derive *composite classes* by tracing the labels through the WordNet hierarchy. We simulate rounds of active learning, starting with only unlabeled data, and proceeding until the dataset is fully labeled, at which point all models achieve the same accuracy. We compare models by their accuracy on fully annotated i.i.d. hold-out data subject to fixed annotation budgets.

Our experiments show that on several datasets, the ERC sampling performs best. On *TinyImageNet*, ERC strictly dominates, improving accuracy by 26% (relative) and 8.1% (absolute) over the supervised baseline with a budget of 250k binary questions. Additionally, ERC can fully annotate the dataset with just 491k binary questions and EDC can do with just 482k binary questions (vs 827k). We make the surprising observation that taking disparate annotation costs into account may alter the conventional wisdom that active learners should solicit labels for *hard* examples. In the ALPF setting, *easy* examples might yield less information, but require many fewer bits to annotate.

2 Related work

We now review research on active learning and learning from partial labels most relevant to this paper.

Binary identification: Identifying an *exact label* through binary questions can be viewed as a classic binary identification problem (Garey, 1972). The goal in binary identification is to identify an unknown object with a minimal number of binary tests (from a finite, pre-specified set). Hyafil & Rivest (1976) proved that finding the optimal strategy for an arbitrary set of binary tests is NP-complete. Since then, greedy algorithms have been studied. One well-known algorithm is called *binary splitting* (Garey & Graham, 1974; Loveland, 1985), which chooses the binary test that most evenly splits the probability mass of remaining classes. Our work differs from classic binary identification in that (i) we address learning from the partial feedback and (ii) we choose each question by making use of probabilities assigned by our most recent classifier.

Our work builds upon the AL framework, which seeks better predictive models than those trained on equal amounts of i.i.d. data by actively deciding which examples to annotate (Box & Draper, 1987; Cohn et al., 1996; Settles, 2010). Classical approaches select examples for which the current predictor is most uncertain. Multiple notions of uncertainty exist: Dagan & Engelson (1995) chooses examples with *maximum entropy* (ME) predictive distributions, while Culotta & McCallum (2005) uses the *least confidence* (LC) heuristic, also referred to as *variation ratios* (Freeman, 1965) sorts examples in ascending order by the probability assigned to the argmax of the predictive distribution. Settles et al. (2008) notes that real annotation costs may vary across data points suggesting cost-aware sampling heuristics. However, Settles et al. (2008) considers only fixed costs per example. In contrast, we decompose the labeling process into the multiple questions it often requires, and use our model to choose the questions to ask. In our setup, labeling costs vary both across examples and over time. Generally, as the learner improves, labeling costs (per example) decrease.

Under restricted settings, AL enjoys theoretical benefits over passive labeling but in general scenarios, the worst-case performance of AL is no better than passive labeling Balcan & Urner (2016). Deep Active Learning (DAL) has recently emerged as an active research area. Notably Wang et al. (2016) explores a scheme that combines traditional heuristics with pseudo-labeling of confidently predicted examples. Gal et al. (2017) notes that the softmax outputs of neural networks do not capture epistemic uncertainty (Kendall & Gal, 2017), proposing instead to use Monte Carlo samples from a dropout-regularized neural network to produce uncertainty estimates. Their strategy annotates examples for which the sampled outputs have maximum disagreement. DAL has also risen in the NLP community, where for many tasks, researchers must contend with small amounts of labeled data. Zhang et al. (2017) explores AL for sentiment classification. They propose a new neural network-based sampling heuristic, choosing examples for which the expected update to the word embeddings is largest. Recently, Shen et al. (2017) proposed DAL for named entity recognition, matching state of the art predictive performance with just 25% of the training data. Kampffmeyer et al. (2016) and Kendall et al. (2015) explore other measures of uncertainty over neural network predictions, but do not address active learning.

Many papers on learning from partial labels (Grandvalet & Bengio, 2004; Nguyen & Caruana, 2008; Cour et al., 2011) assume that partial labels are given a priori and fixed. Grandvalet & Bengio (2004) formalizes the partial labeling problem in the probabilistic framework and proposes a minimum entropy based solution. Nguyen & Caruana (2008) proposes an efficient algorithm to learn classifiers from partial labels within the max-margin framework. Cour et al. (2011) addresses desirable properties of partial labels that allow us to learn from them effectively.

Key differences As in Grandvalet & Bengio (2004), we maximize the log probability of the partial label, calculated by marginalizing over the atomic classes it contains. However, we do not use a regularizer to encourage non-uniform distribution among predicted probabilities on partial labels. In contrast to these papers, which assume a fixed set of available partial labels, we *actively* solicit partial feedback. This presents new algorithmic challenges: (i) the partial labels each data point changes across training rounds; (ii) the partial labels result from an active selection, which introduces bias; and (iii) our problem setup requires a sampling strategy to choose which composite class to query.

3 Active Learning with Partial Feedback

3.1 Formulation

This work addresses the multiclass classification setting. By $\boldsymbol{x} \in \mathcal{R}^d$ and $y \in \mathcal{Y}$ for $\mathcal{Y} = \{\{1\}, ..., \{k\}\}$, we denote feature vectors and labels. Here d is the feature dimension and k is the number of *atomic* classes. By *atomic* class, we mean that they are indivisible. As in conventional AL, the agent starts off with an unlabeled training set $\mathcal{D} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$.

Composite classes We also consider a pre-specified collection of composite classes $\mathcal{C} = \{c_1, ..., c_m\}$, where each composite class $c_i \subset \{1, ..., k\}$ is a subset of labels such that $|c_i| \geq 1$. Note that $\mathcal{C}$ includes both the atomic and composite classes. In this paper's empirical section, we generate composite classes by imposing an existing lexical hierarchy on the class labels (Miller, 1995).

Partial labels For an example i, we use *partial label* to describe any element $\tilde{y}_i \subset \{1, ..., k\}$ such that $\tilde{y}_i \supset y_i$. We call $\tilde{y}_i$ a *partial label* because it may rule out some classes, but doesn't fully indicate underlying atomic class. For example, *dog* $= \{$*akita, beagle, bulldog, ...*$\}$ is a valid partial label when the true label is *bulldog*. An ALPF learner eliminates classes, obtaining successively smaller partial labels, until only one (the *exact label*) remains. To simplify notation, in this paper, by an example's *partial label*, we refer to the smallest partial label available based on the already-eliminated classes. At any step t and for any example i, we use $\tilde{y}_i^{(t)}$ to denote the current partial label. The initial partial label for every example is $\tilde{y}^0 = \{1, ..., k\}$ An *exact label* is achieved when the partial label $\tilde{y}_i = y_i$.

Partial Feedback The set of possible questions $\mathcal{Q} = \mathcal{X} \times \mathcal{C}$ includes all pairs of examples and composite classes. An ALPF learner interacts with an annotator by choosing a question $q \in \mathcal{Q}$. Informally, we pick a question $q = (\boldsymbol{x}_i, c_j)$ and ask the annotator, *does $\boldsymbol{x}_i$ contain a c_j?* If the queried example's label belongs to the queried composite class ($y_i \subset c_j$), the answer is 1, else 0.

Let α_q denote the binary answer to question $q \in \mathcal{Q}$. Based on the partial feedback, we can compute the new partial label $\tilde{y}^{(t+1)}$ according to Eq. (1),

$$\tilde{y}^{(t+1)} = \begin{cases} \tilde{y}^{(t)} \setminus c & \text{if } \alpha = 0 \\ \tilde{y}^{(t)} \setminus \overline{c} & \text{if } \alpha = 1 \end{cases} \tag{1}$$

Note that here $\tilde{y}^{(t)}$ and c are sets, α is a bit, $\overline{c}$ is a set complement, and that $\tilde{y}^{(t)} \setminus \overline{c}$ and $\tilde{y}^{(t)} \setminus c$ are set subtractions to eliminate (formerly potential) classes from the partial label based on the answer.

Learning Process The learning process is simple: At each round t, the learner selects a pair $(\boldsymbol{x}, c)$ for labeling. Note that a rational agent will never select either (i) an example for which the exact label is known, or (ii) a pair $(\boldsymbol{x}, c)$ for which the answer is already known, e.g., if $c \supset \tilde{y}^{(t)}$ or $c \cap \tilde{y}^{(t)} = \emptyset$. After receiving binary feedback, the agent updates the corresponding partial label $\tilde{y}^{(t)} \rightarrow \tilde{y}^{(t+1)}$, using Equation 1. The agent then re-estimates its model, using all available non-trivial partial labels and selects another question q. In batch-mode, the ALPF learner re-estimates its model once per T rounds which is necessary when training is expensive (e.g. deep learning).

Objectives We state two goals for ALPF learners. First, we want predictors with **low error on exactly labeled iid holdout data**, for any given setting of a fixed annotation budget. Second, we want to **fully annotate datasets at the lowest cost**. In our experiments (Section 4), the best ALPF strategy dominates on both tasks.

3.2 Learning from partial labels

We now address the task of learning a multiclass classifier from partial labels, a fundamental requirement of ALPF, regardless of the choice of sampling strategy. At time t, our model $\hat{y}(y, \boldsymbol{x}, \theta^{(t)})$ parameterised by parameters $\theta^{(t)}$ estimates the conditional probability of an atomic class y. For simplicity, when the context is clear, we will use $\hat{\boldsymbol{y}}$ to designate the full vector of predicted probabilities over all classes. The probability assigned to a partial label $\tilde{y}$ can be expressed by marginalizing over the atomic classes that it contains as follows:

$$\hat{p}(\tilde{y}^{(t)}, \boldsymbol{x}, \theta^{(t)}) = \sum_{y \in \tilde{y}^{(t)}} \hat{y}(y, \boldsymbol{x}, \theta^{(t)}).$$

To optimize our model, we minimize the negative log probability assigned to the following loss:

$$\mathcal{L}(\theta^{(t+1)}) = -\frac{1}{n} \sum_{i=1}^{n} \log \left[\hat{p}(\tilde{y}_i^{(t)}, \boldsymbol{x}_i, \theta^{(t)}) \right] \tag{2}$$

Note that when every example is exactly labeled, our loss function simplifies to the standard cross entropy loss often used for multi-class classification. Also note that when every partial label contains the full set of classes, all partial labels have probability 1 and the update is a no-op. Finally, if the partial label indicates a composite class such as *dog*, and the predictive probability mass is exclusively allocated among various breeds of dog, our loss will be 0. Models are only updated when their predictions disagree (to some degree) with the current partial label.

3.3 Sampling strategies

This section introduces introduces three sampling strategies for choosing (example, question) pairs.

Expected Information Gain (EIG): Following previous work on active learning, for any example x, we can quantify the classifer's uncertainty as the entropy of the predictive distribution. In the AL setting, each query returns an exact label, and thus the post-query entropy is always 0. There, the information gained is precisely the entropy of $\hat{y}$. In our case, each query yields a different partial label depending upon the answer. We use the notation $\hat{\boldsymbol{y}}_0$, and $\hat{\boldsymbol{y}}_1$ denote a modified predicted distribution, after eliminating classes ruled out under the new partial label and normalizing the remaining $\hat{y}_i$ to sum to 1. We generalize the maximum

Algorithm 1 Active Learning with Partial Feedback

```
**Input:** $\mathcal{D} \leftarrow [\boldsymbol{x}_i]_{i=1}^N, \mathcal{C} \leftarrow [c_j]_{j=1}^M, k, T$
**Initialize:** $\tilde{y}_i^{(0)} \leftarrow \{1, \ldots, k\}, \theta \leftarrow \theta^{(0)}, t \leftarrow 0$
**repeat**
  Score every $(\boldsymbol{x}_i, c_j)$ with $\theta$
  **repeat**
    Select $(\boldsymbol{x}_{i^*}, c_{j^*})$ with the best score
    Query $c_{j^*}$ on data $\boldsymbol{x}_{i^*}$
    Receive feedback $\alpha$
    Update $\tilde{y}_{i^*}^{(t+1)}$ according to $\alpha$
    $t \leftarrow t + 1$
  **until** $(t \bmod T = 0)$ or $(\forall i, |\tilde{y}_i^{(t)}| = 1)$
  $\theta \leftarrow \arg\min_\theta \mathcal{L}(\theta)$
**until** $\forall i, |\tilde{y}_i^{(t)}| = 1$ or $t$ exhausts budget
```

entropy criteria to ALPF by selecting those examples for which the expected reduction in entropy (based on the current predictions) is greatest.

$$EIG_{(\boldsymbol{x},c)} = S(\hat{\boldsymbol{y}}) - [\hat{p}(c, \boldsymbol{x}, \theta) S(\hat{\boldsymbol{y}}_1) + (1 - \hat{p}(c, \boldsymbol{x}, \theta)) S(\hat{\boldsymbol{y}}_0)] \tag{3}$$

where $S(\cdot)$ is the entropy of a probability mass function. It is easy to prove that EIG is maximized when $\hat{p}(c, \boldsymbol{x}, \theta) = 0.5$. Put informally, EIG favors binary questions for which the answer is most uncertain. This is consistent with the greedy binary splitting strategy in the binary identification problem.

Expected Remaining Classes (ERC): Next, we propose ERC, a heuristic that suggests arriving as quickly as possible at exactly-labeled examples. At each round, ERC selects those examples for which the expected number of remaining classes is fewest:

$$ERC_{(\boldsymbol{x},c)} = \hat{p}(c, \boldsymbol{x}, \theta) ||\hat{\boldsymbol{y}}_1||_0 + (1 - \hat{p}(c, \boldsymbol{x}, \theta)) ||\hat{\boldsymbol{y}}_0||_0, \tag{4}$$

where $||\hat{\boldsymbol{y}}_\alpha||$ is the size of the partial label following given answer α. ERC is minimized when the result of the feedback will produce an exact label with probability 1. For a given example $\boldsymbol{x}_i$, if $||\hat{\boldsymbol{y}}_i||_0 = 2$, containing only the potential classes (e.g.) *dog* and *cat*, then with certainty, ERC will produce an exact label by querying the class $\{$*dog*$\}$ (or equivalently $\{$*cat*$\}$). This is inspired by Cour et al. (2011), which shows that the partial classification loss (what we optimize with partial labels) is an upper bound of the true classification loss (as if true labels are available) with a linear factor of $\frac{1}{1-\varepsilon}$, where ε is ambiguity degree and $\varepsilon \propto |\tilde{y}|$. By selecting $q \in \mathcal{Q}$ that leads to the smallest $|\tilde{y}|$, we can tighten the bound to make optimization with partial labels more effective.

Expected Decrease in Classes (EDC): More in keeping with the traditional goal of minimizing uncertainty, we might choose EDC, the sampling strategy which we expect to result in the greatest reduction in the number of potential classes. We can express EDC as the difference between the number of potential labels (known) and the expected number of potential labels remaining (expressed by ERC).

$$EDC_{(\boldsymbol{x},c)} = |\tilde{y}^{(t)}| - ERC_{(\boldsymbol{x},c)} \tag{5}$$

Algorithm: We summarize the workflow of a ALPF learner in Algorithm 1.

4 Experiments

4.1 Experimental Setup

We evaluate ALPF algorithms on the CIFAR10, CIFAR100, and Tiny ImageNet datasets, with training sets of 50k, 50k, and 100k examples, and 10, 100, and 200 classes respectively. After imposing the Wordnet hierarchy on the label names, the size of the set of possible binary questions $|\mathcal{C}|$ for each dataset are 42, 334, and 346, respectively. The number of binary questions between re-trainings are 5k, 15k, and 30k, respectively. By default, we warm-start each learner with the same 5% of training examples selected i.i.d. and exactly labeled. Warm-starting has proven essential in other papers combining deep and active learning Shen et al. (2017). **Intuition:** given just a few labels, a deep neural network will over-fit, becoming badly uncalibrated. At this point, uncertainty estimates become unreliable and common sampling strategies cease to work.

Model For each experiment, the underlying classifier adopts the ResNet-18 architecture He et al. (2016). We initialize weights with the *Xavier* technique (Glorot & Bengio, 2010) and minimize our loss using the Adam Kingma & Ba (2014) optimizer. Notably, we found that Adam outperforms SGD significantly when learning from partial labels. Each experiment uses an initial learning rate of 0.001, first-order momentum decay (β_1) of 0.9, and second-order momentum decay (β_2) of 0.999. Finally, we train with mini-batches of 200 examples and perform standard data augmentation techniques including random cropping, resizing, and mirror-flipping. We implement all models in MXNet and commit to open-sourcing our code upon publication.

Re-training Ideally, we would update our model immediately after receiving feedback, but for deep neural networks, this is too expensive. We run all experiments starting from 5% labeled data and continue each experiment until every example is exactly labeled, at which point all strategies produce the same accuracy. At each round, we re-train our classifier from scratch with random initialization. Alternatively, we could initialize the new classifier with the previous best one (as in Shen et al. (2017)). However, preliminary experiments showed that while this yields faster convergence, we get considerably worse performance, perhaps owing to greater over-fitting on those labels acquired early in training. In all experiments, for simplicity, we terminate the optimization after 75 epochs.

4.2 Learning from partial labels

Since the success of ALPF depends in part on learning from partial labels, we first demonstrate the efficacy of learning from partial labels with our loss function when the partial labels are given a priori. In these experiments we simulate a partially labeled dataset and show that the learner achieves significantly better accuracy when learning from partial labels than if it excluded the partial labels and focused only on exactly annotated examples. Using our WordNet-derived hierarchy, we conduct experiments with partial labels at different levels of granularity. Using partial labels from one level above the leaf, *German shepherd* becomes *dog*. Going up two levels in the hierarchy, we get *animal*.

We first train a standard multi-class classifier with γ (%) *exactly labeled* training data. Then we train another classifier with an additional $(1-\gamma)$ that is *partially labeled* in a different granularity (level of hierarchy). We compare the classifier performance on holdout data both *with* and *without* adding *partial labels* in Table 1. We make two key observations: (i) additional coarse-grained partial labels improve model accuracy (ii) as

Table 1: Learning from partial labels on Tiny ImageNet

$\gamma(\%)$	γ	$+(1-\gamma)$		
	$\mathcal{Y}$	$\tilde{\mathcal{Y}}_1$	$\tilde{\mathcal{Y}}_2$	$\tilde{\mathcal{Y}}_4$
20	0.285	**+0.113**	+0.086	+0.025
40	0.351	+0.079	+0.056	+0.016
60	0.391	+0.051	+0.036	+0.018
80	0.432	+0.015	+0.017	-0.009
100	0.441	-	-	-

expected, the improvement diminishes as partial label gets coarser. These observations suggest that our loss function enables effective learning from partial labels.

4.3 Sampling strategies

Baseline Our baseline learner samples examples at random from a uniform distribution. Once an example is sampled, the learner applies top-down binary splitting with equal probabilities to sample questions from the hierarchy until the example is exactly labeled. Note that this sampling strategy is equivalent to applying EIG with uniform $\hat{y}$. We will refer to *binary splitting* with equal probabilities as *passive binary splitting* in short.

AL We compare to two conventional AL approaches, which combine uncertainty sampling with *binary splitting*. At each round, the AL learner chooses the example according to either the ME or LC heuristic. After choosing the example, the AL learner follows the passive binary splitting to sample questions until *exactly label* the example.

AQ We also create three *active questions* learners, which choose examples at random, but then solicit partial feedback using EIG, EDC, or ERC. AQ models only move on to the next example once finding the current example's exact label.

ALPF We introduce active learners that make use of partial feedback. Now we actively choose both the next example and question according to the sampling strategies. Importantly, now it may choose another example even if the current one has **not** been *exactly labeled*. Therefore, classifier may encounter partial labels during training, while for all three above the learning is still fully supervised.

Results We run all methods above until fully annotate the training set on CIFAR10, CIFAR100, Tiny ImageNet. We evaluate each method from two perspectives: learning and annotation. For learning, we evaluate model performance (i.e. multi-class classification accuracy on an *exactly labeled* i.i.d. holdout set), subject to a fixed annotation budget. In terms of annotation, we evaluate the total number of binary questions each learner asks before *exactly labeling* all training examples. We compile our results in Table 2, in which labeling costs are rounded to 10%, 20% etc. for ease of presentation and budget includes the (5%) cost of warm-starting with i.i.d. labels. We highlight two observations:

(1) AL methods do not improve upon baseline. Since neither AL or baseline employ active questions, each example costs the same for both to obtain *exact feedback*. The lack of improvement suggests that

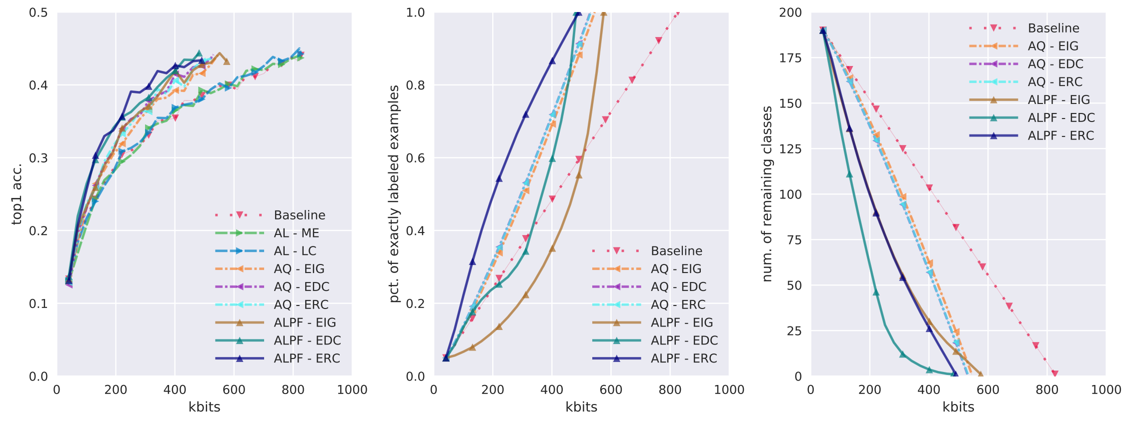


Figure 2: The progression of top1 classification accuracy (left), percentage of exactly labeled training examples (middle), and average number of remaining classes (right).

uncertainty sampling is not well-suited for this task (i.e. multi-class image classification with DNNs). We hypothesize that current optimizers for deep learning (e.g., stochastic gradient descent) prefer i.i.d data rather than actively-sampled batches.

(2) Given the above results, it is natural to sample random data but actively choose binary questions until *exact feedback* is provided for each datum (AQ). AQ provides a stark improvement over the baseline, and *moreover, the improvement grows with more annotation*. On *TinyImageNet*, the gap between the best AQ and baseline is 1.0% at 10% binary annotations. Then at 50%, the gap becomes 6.1%. Finally, allowing the active learner to take advantage of *partial feedback* (ALPF) further improves overall learning efficiency, outperforming AQ methods by up to 3.9% (i.e. at 20%). The best ALPF method (i.e. ALPF-ERC) outperforms the baseline approach by up to 8.1% at 30%. This evidence suggests only partial feedback may not only be a viable annotation strategy but also a more efficient one.

4.4 Diagnostic analyses

We now present a few diagnostic analyses. First, we study how different amounts of warm-starting affects ALPF learners' performance with a small set of i.i.d. labels. Second, we compare the selections due to ERC and EDC to those produced through uncertainty sampling. Third, we note that while EDC and ERC appear to perform best on our problems, they may be vulnerable to over-focusing on easy classes given a large dataset, small budget and when a few classes are trivial to recognize. We examine this setting by creating an adversarial dataset intended to break the heuristics.

Warm-starting We compare the performance of each strategy under different percentages (0%, 5%, and 10%) of pre-labeled i.i.d. data. We visualize the results in Figure 5 (Appendix). Results show that ERC works properly even without warm-starting, while EIG benefits from a 5% warm-start and EDC suffers badly without warm-starting. We observe that 10% warm-starting yields no further improvement.

Table 2: Results on Tiny ImageNet

	Annotation Budget						Labeling Cost
	10%	20%	30%	40%	50%	100%	
TinyImageNet							
Baseline	0.186	0.266	0.310	0.351	0.354	0.441	827k
AL - ME	0.169	0.269	0.303	0.347	0.365	-	827k
AL - LC	0.184	0.262	0.313	0.355	0.369	-	827k
AQ - EIG	0.186	0.283	0.336	0.381	0.393	-	545k
AQ - EDC	0.196	0.291	0.353	0.386	0.415	-	530k
AQ - ERC	0.194	0.295	0.346	0.394	0.406	-	531k
ALPF - EIG	0.203	0.289	0.351	0.384	0.420	-	575k
ALPF - EDC	**0.220**	0.319	0.363	0.397	0.420	-	482k
ALPF - ERC	0.207	**0.330**	**0.391**	**0.419**	**0.427**	-	491k
CIFAR100							
Baseline	0.252	0.340	0.412	0.437	0.469	0.537	337k
ALPF - EIG	0.263	0.341	0.423	0.466	0.497	-	235k
ALPF - EDC	**0.281**	0.367	0.442	0.479	0.518	-	193k
ALPF - ERC	0.273	**0.379**	**0.464**	**0.502**	**0.526**	-	187k
CIFAR10							
Baseline	0.645	0.718	0.757	0.778	0.792	0.829	170k
ALPF - EIG	0.673	0.741	0.786	0.815	0.813	-	124k
ALPF - EDC	**0.676**	**0.752**	**0.797**	0.832	N/A	-	74k
ALPF - ERC	0.670	0.743	**0.797**	**0.833**	N/A	-	74k

Sample uncertainty Classic uncertainty sampling chooses data of high uncertainty. This question is worth reexamining in the context of ALPF. To analyze the behavior of ALPF learners vis-a-vis uncertainty we plot average prediction entropy of sampled data for ALPF learners with different sampling strategies (Figure 3). Note that ALPF learners using EIG pick high-entropy data, while ALPF learners with EDC and ERC choose examples with lower entropy predictions. The (perhaps) surprising performance of EDC and ERC may owe to the cost structure of ALPF. While labels for examples with low-entropy predictions confer less information, they also come at lower cost.

Adversarial setting Because ERC goes after easy examples, we test its behavior on a simulated dataset in which a subset of classes are trivially easy. We randomly pick 2 out of 10 classes on *CIFAR10* and for one, set all pixels white, and for the other, set all pixels black. We plot the label distribution of the selected data in Figure 4. We also plot the sampled label distribution on unperturbed *CIFAR10*. Light green and light purple represent the two classes made artificially easy. As we can see, in the normal case, EIG splits budget among all classes roughly evenly while EDC and ERC focus more on different classes at different stages. In the adversarial case, EIG only starts querying easy images until it classifiers those classes confidently while EDC and ERC concentrate on exhausting the easy ones. Although EDC and ERC still manage to label all

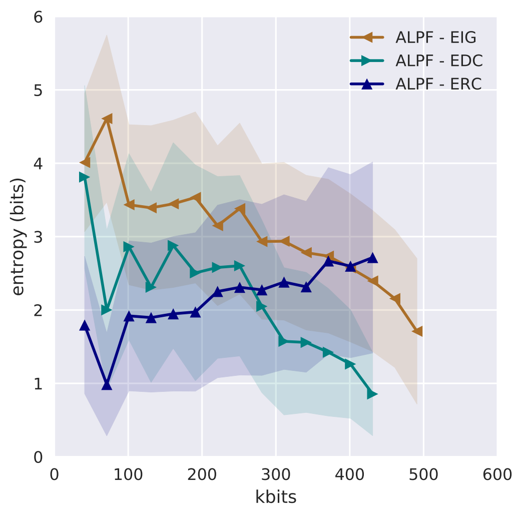


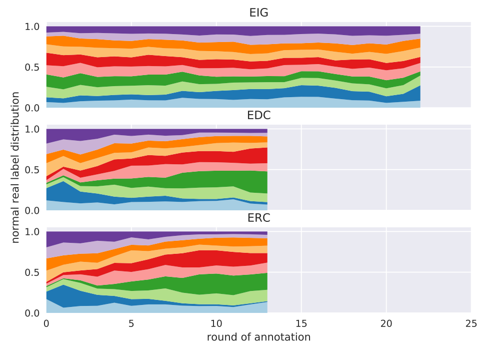
Figure 3: Analysis of samples' uncertainty.

data with less total cost than EIG, this behavior might cost us when we have (i) trivial classes, (ii) enormous unlabeled dataset, and (iii) a limited budget. In these cases ERC might spend its whole budget on the easy examples.

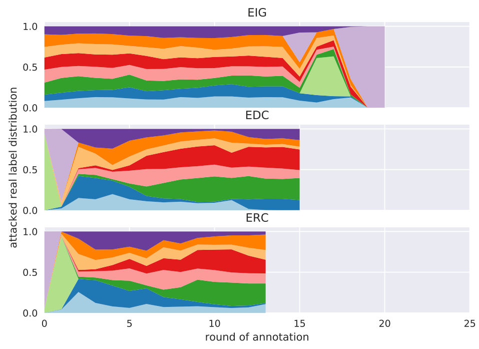
5 Conclusion

Traditional active learners treat human annotation as a black box. Because of this, human annotators often overwork for complicated labeling tasks and end up spending time less efficiently. We develop an active learning framework, resembling the classic binary identification problem, that breaks down complicated labeling into choosing a sequence of binary questions, allowing annotators to provide partial feedback. Our experiments validate the active learning with partial feedback framework on large-scale classification benchmarks. We find that the best among our proposed ALPF learners achieves maximum performance and fully labels the data with 42% less binary questions, comparing to passive learners and traditional active learners. Our diagnostic analysis suggests that in ALPF, it is sometimes more efficient to start with "easier" examples that can be cheaply annotated rather than with "harder" data as often suggested by traditional active learning.

Future work: We hope to apply APLF to other supervised tasks, such as multi-label classification, where annotators may assign multiple labels to each data item. We also wish to apply APLF to construct next-generation annotated datasets.



(a) Normal CIFAR10



(b) Adversarial CIFAR10

Figure 4: Analysis on samples' real label distribution.

References

Balcan, M.-F., & Urner, R. (2016). Active learning–modern learning theory. *Encyclopedia of Algorithms*.

Box, G. E., & Draper, N. R. (1987). *Empirical model-building and response surfaces.*. John Wiley & Sons.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of artificial intelligence research*.

Cour, T., Sapp, B., & Taskar, B. (2011). Learning from partial labels. *Journal of Machine Learning Research*, *12*(May), 1501–1536.

Culotta, A., & McCallum, A. (2005). Reducing labeling effort for structured prediction tasks.

Dagan, I., & Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers.

Freeman, L. C. (1965). *Elementary applied statistics: for students in behavioral science*. John Wiley & Sons.

Gal, Y., Islam, R., & Ghahramani, Z. (2017). Deep bayesian active learning with image data. *arXiv:1703.02910*.

Garey, M. R. (1972). Optimal binary identification procedures. *SIAM Journal on Applied Mathematics*.

Garey, M. R., & Graham, R. L. (1974). Performance bounds on the splitting algorithm for binary testing. *Acta Informatica*.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, (pp. 249–256).

Grandvalet, Y., & Bengio, Y. (2004). Learning from partial labels with minimum entropy.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 770–778).

Hyafil, L., & Rivest, R. L. (1976). Constructing optimal binary decision trees is np-complete. *Information processing letters*.

Kakade, S. M., Shalev-Shwartz, S., & Tewari, A. (2008). Efficient bandit algorithms for online multiclass prediction. In *Proceedings of the 25th international conference on Machine learning*, (pp. 440–447). ACM.

Kampffmeyer, M., Salberg, A.-B., & Jenssen, R. (2016). Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In *CVPR*.

Kendall, A., Badrinarayanan, V., & Cipolla, R. (2015). Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv:1511.02680*.

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*.

Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lipscomb, C. E. (2000). Medical subject headings (mesh). *Bulletin of the Medical Library Association*, *88*(3), 265.

Loveland, D. W. (1985). Performance bounds for binary testing with arbitrary weights. *Acta Informatica*.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, *38*(11), 39–41.

Nguyen, N., & Caruana, R. (2008). Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 551–559). ACM.

Settles, B. (2010). Active learning literature survey. *University of Wisconsin, Madison*, *52*(55-66), 11.

Settles, B. (2011). From theories to queries: Active learning in practice. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, (pp. 1–18).

Settles, B., Craven, M., & Friedland, L. (2008). Active learning with real annotation costs.

Shen, Y., Yun, H., Lipton, Z. C., Kronrod, Y., & Anandkumar, A. (2017). Deep active learning for named entity recognition. *arXiv preprint arXiv:1707.05928*.

Wang, K., Zhang, D., Li, Y., Zhang, R., & Lin, L. (2016). Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*.

Zhang, Y., Lease, M., & Wallace, B. C. (2017). Active discriminative text representation learning.

A Warm-starting Plot

In this brief appendix, find a plot comparing our strategies under various amounts of warm-starting with pre-labeled i.i.d. data.

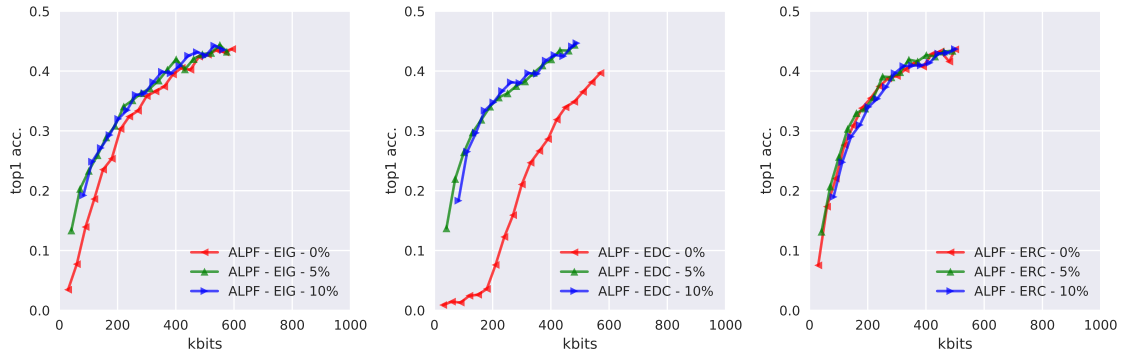


Figure 5: This plot compares our models under various amounts of warm-starting with pre-labeled i.i.d. data. We find that on the investigated datasets, ERC does benefit from warm-starting. However, absent warm-starting, EIG performs significantly worse and EDC suffers even more. We find that 5% warmstarting helps these two models and that for both, increasing warm-starting from 5% up to 10% does not lead to further improvements.