To appear, Proceedings of the International Conference on Computer Vision (ICCV), 2011.

Actively Selecting Annotations Among Objects and Attributes

Adriana Kovashka Sudheendra Vijayanarasimhan Kristen Grauman
University of Texas at Austin
{adriana, svnaras, grauman}@cs.utexas.edu

Abstract

We present an active learning approach to choose image annotation requests among both object category labels and the objects' attribute labels. The goal is to solicit those labels that will best use human effort when training a multi-class object recognition model. In contrast to previous work in active visual category learning, our approach directly exploits the dependencies between human-nameable visual attributes and the objects they describe, shifting its requests in either label space accordingly. We adopt a discriminative latent model that captures object-attribute and attribute-attribute relationships, and then define a suitable entropy reduction selection criterion to predict the influence a new label might have throughout those connections. On three challenging datasets, we demonstrate that the method can more successfully accelerate object learning relative to both passive learning and traditional active learning approaches.

1. Introduction

Many state-of-the-art object recognition systems integrate robust visual descriptors with a supervised learning algorithm. This basic framework entails having humans "teach" the machine learner about objects through labeled examples, which makes the data collection process itself of critical importance. As such, recent research explores interesting issues in gathering large datasets of Web images [21, 24, 10, 3], mining external knowledge sources [19, 1, 2], creating benchmark challenges [7], and developing new methods to reduce the expense of manual annotations. Active learning methods in particular are a promising way to focus human effort, as the system can request labels only for those instances that appear most informative based on its current category models [17, 25, 26, 13, 12, 23].

In spite of such progress, however, substantial challenges remain. First of all, most existing techniques assume that the labels of interest are the object category names, yet recent work shows the need to move "beyond labels" to even richer annotations such as descriptive *attributes* or relationships between objects [15, 14, 6, 29, 10, 11]. Attributes are high-level features that describe traits of an object such as physical properties, behavior, or uses; for example, while object labels might include *house*, *phone*, and *dog*, attribute labels might include *wooden*, *furry*, or *red*. Secondly, real-world applications of object recognition demand scaling to a very large number of categories, which at the surface suggests that the number of labels needed must grow proportionally with the number of total classes considered—even if one plans to collect labels with active learning.

We propose an active learning approach to address these issues. The main idea is to actively select image annotation requests among *both* object category labels as well as the objects' shared attributes, so as to acquire the labels expected to most reduce total uncertainty for multi-class ob-

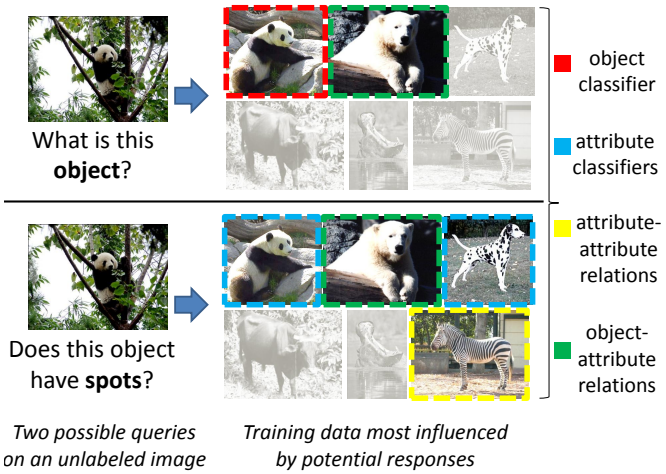

Figure 1. Object and attribute labels affect the current model's understanding of each training image in distinct ways. This example illustrates how the different label requests about the image (left) will influence the different components of the learned models (right, color coded by type of impact). For example, whereas getting the 'panda' label may reduce uncertainty about that class and refine the model's distinctions with other bear classes (top), getting the 'spotted' label could have even greater influence, strengthening discriminability for the striped and spotted attributes alike.

ject predictions. This means, for example, that during one active learning loop a human may be asked to name an object, whereas in the next s/he may be asked to state whether a particular attribute is present. The goal is to select those pairs of images and labeling questions that will be most useful given the current models.

By simultaneously weighing requests in both label spaces, we expect the learner to more efficiently refine its object models. Why? Knowledge of an attribute's presence in an image can immediately influence many object models, since attributes are by definition shared across subsets of the object categories. At the same time, attributes' presence or absence in an image is often correlated (e.g., if something "has skin" it is unlikely to be "metallic" as well), suggesting that many images do not require a full annotation of all attributes.[1] See Fig. 1.

To implement the proposed idea, we adopt a discriminative latent model [29] that captures object-attribute and attribute-attribute relationships, and then define a suitable entropy reduction selection criterion to predict the influence a new label of either type will have throughout those connections. This criterion estimates the expected entropy change on all labeled and unlabeled examples, should the label under consideration be obtained. We adapt the existing classifier to extract the necessary posterior probability estimates, and show how to handle partially labeled examples (i.e., those with only some attributes known) such that they can have immediate influence on the active selection.

A novel aspect of our approach is that it both weighs different annotation requests and also models dependencies within multi-label instances. Only limited prior work explores either one or the other aspect [25, 23, 17], and in a different context than our setting here. Furthermore, in contrast to any existing active learning work, our approach exploits dependencies between the target label space and a latent but human-describable label space, and is the first to learn objects and attributes actively in concert. This can also be viewed as a new way to efficiently supervise joint multi-class training, in that the actively selected attribute labeling questions are directly tied to properties shared across classes.

2. Related Work

Recent work explores several ways to use visual attributes in object recognition. Since attributes are shared across categories, they enable knowledge transfer to recognize unseen objects [15] and describe novel instances [8]. By integrating the learning process for both objects and attributes, one can use weak supervision more effectively [27] and even improve object recognition accuracy [14, 29]. In addition, capturing the relationships *between* attributes can strengthen object category models, as first shown by Wang and Mori [29]. We employ their latent discriminative model for classification, as it suitably represents all the object and attribute interactions of interest to our active learning approach.

Most work using attributes assumes that images are fully labeled with all their attributes, either through a top-down labeling of the object classes (e.g., all bears are 'furry' [15]) or by individually providing attributes for each image [8, 6]. To alleviate this burden, researchers study ways to learn attribute classifiers from noisy keyword search data [10], or to automatically discover the attributes and objects' semantic relatedness from Web images and text sources [1, 19]. In contrast to these unsupervised methods, our work explicitly engages a human annotator to respond to object or attribute queries where most needed.

Active learning for object recognition typically reduces human labeling effort by selecting the most uncertain exemplar to get labeled with its object category name(s) [17, 31, 13, 12]. More closely related to our approach, some work further shows how to actively integrate annotations of different *levels*, i.e., by alternately requesting segmented regions or asking about the contextual relationships between objects in an image [25, 26, 23]. In the realm of natural language processing, researchers develop ways to actively ask humans which words may be relevant for a document classification task [18, 5]; words could be seen as a loose analogue for attributes, though we do not consider requests about visual attribute relevance.

Active visual learning methods generally do not account for the dependencies between labels on the same image. An exception is the scene classification method of [17], which learns with multi-label images and requests the most informative image-label pair. However, its selection strategy considers only the local effects of a candidate label request, by measuring the uncertainty and label correlations for each individual image in isolation. In contrast, the proposed selection method evaluates the influence of the candidate label if propagated to all current models, which is critical to achieve our goal of exploiting shared latent attributes to reduce annotation effort.

While the above work tackles active *learning*, a system for active *classification* is developed in [2], where the system interactively deduces the object label for a single novel image by asking a human to label a sequence of its attributes. In contrast, our method uses the human annotators during the iterative training process for all object categories, and then makes predictions on novel images without human intervention. Furthermore, our method requests information from the annotators on two levels (object and attribute labels), whereas the method in [2] only requests attribute labels.

[1] In fact, blindly requesting all attributes may not only be wasteful, but it may also be impractical at the interface level once we consider large attribute vocabularies of hundreds or more properties.

3. Approach

The proposed approach requires two main elements: a unified classification model to capture object and attribute relationships, and a way to weigh candidate requests for either label space. We first briefly describe the classifier (Sec. 3.1), and then explain how to actively improve it with an entropy-based selection criterion (Sec. 3.2).

3.1. Object-Attribute Model

In order to predict the impact of potential object and attribute label requests, we need a classifier that accounts for all four relationships portrayed in Fig. 1. To this end, we directly adopt the discriminative model recently proposed by Wang and Mori [29]. We briefly summarize the necessary background in this section; see [29] for details.

The model is a multi-class *object classifier* that uses attributes as hidden variables. The relationships between the object categories and the attributes are learned parameters in the model. Relationships between attributes are represented in a tree-structured graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose vertices denote the K attributes and whose edges are restricted to pairs of attributes $(j, k) \in \mathcal{E}$ that have the highest mutual information[2]; parameters reflecting the importance of those dependencies for distinguishing objects are then incorporated into the main classifier.

A fully labeled training example consists of an image $\mathbf{x} \in \mathcal{X}$, its object label $y \in \mathcal{Y}$, and an indicator vector of K attribute labels $\mathbf{h} = [h_1, \ldots, h_K]$, with all $h_i \in \mathcal{A}$. We use binary-valued attributes, and so $\mathcal{A} = \{0, 1\}$. The classifier $f_{\mathbf{w}} : \mathcal{X} \times \mathcal{Y} \rightarrow \Re$ is parameterized by vector $\mathbf{w}$, and will be defined below. At test time, one predicts the object label y^* for image $\mathbf{x}$ as:

$$y^* = \arg\max_{y \in \mathcal{Y}} f_{\mathbf{w}}(\mathbf{x}, y), \tag{1}$$

where, following the general latent SVM approach [9, 28], the discriminant function is maximized over all possible latent attribute label assignments:

$$f_{\mathbf{w}}(\mathbf{x}, y) = \max_{\mathbf{h}} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}, y), \tag{2}$$

where $\Phi(\mathbf{x}, \mathbf{h}, y)$ is a feature vector that depends on its arguments.

Wang and Mori define the model as follows[3]:

$$\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}, y) = \mathbf{w}_y^T \phi(\mathbf{x}; y) + \sum_{j \in \mathcal{V}} \mathbf{w}_{h_j}^T \varphi(\mathbf{x}; j, h_j)$$

$$+ \sum_{(j,k) \in \mathcal{E}} \mathbf{w}_{j,k}^T \psi(h_j, h_k) + \sum_{j \in \mathcal{V}} v_{y,h_j}, \tag{3}$$

where $\mathbf{w}$ is the concatenation of all the first parameters appearing in the summands, and the other terms are the features composing $\Phi(\mathbf{x}, \mathbf{h}, y)$.

[2] as found with a maximum spanning tree on a fully connected graph

[3] We omit the class-specific attribute classifier proposed in [29].

Those four features are defined as follows:

- **Object class component:** $\phi(\mathbf{x}; y)$ is the probability image $\mathbf{x}$ has object label y, which is obtained by training a multi-class SVM (ignoring attributes).
- **Attribute class component:** $\varphi(\mathbf{x}; j, h_j)$ is the probability that the j-th attribute is h_j, obtained by training a binary SVM for attribute j (ignoring object labels).
- **Attribute-attribute component:** $\psi(h_j, h_k)$ is a binary vector of length 4, with a 1 in one entry denoting which of the four possibilities is true: that both, neither, the j-th, or the k-th attributes are present.
- **Object-attribute component:** v_{y,h_j} is a learned parameter reflecting the frequency of object being y and the j-th attribute being h_j.

Note that the two first components use separately trained traditional SVMs to produce feature values, which are then weighted by the learned parameters $\mathbf{w}_y$ and $\mathbf{w}_{h_j}$.

To train the model (learn $\mathbf{w}$), we use the non-convex cutting plane method of [4], which allows latent attribute labels for the training examples. We use a mixture of observed and latent attribute labels when dealing with partially labeled examples during the active learning loop (see Sec. 3.2.3). We use the true values of training images' attribute labels when computing the hinge loss function in [29].

During testing, we use linear programming to determine the attribute labels $\mathbf{h}_y^*$ that maximize $f_{\mathbf{w}}$ (for every y), and then predict the object label y^* as in Eqn. 1.

3.2. Active Learning with Objects and Attributes

We take a pool-based active learning approach, where the active learner surveys all unlabeled data to determine which labels to request next. In particular, we use an entropy-based function to score all ⟨image,label request⟩ pairs according to their expected information, and then select the top ranked requests for labeling. Each request asks for one label: the object class, or a specific attribute value. After a human answers the selected requests, the newly labeled instances are appended to the training set (whether for attributes or objects), and then the classifier parameters are updated accordingly. The entire process repeats, for as long as more labeling effort is available. The product is a classifier that predicts object labels. Fig. 2 summarizes the main data flow.

In the following, we first define the sets of annotations that contribute to each component of the model (Sec. 3.2.1). Then we define the active selection function to rank the candidate label requests (Sec. 3.2.2), and then explain how those requests which the system acquires are used to update the model (Sec. 3.2.3).

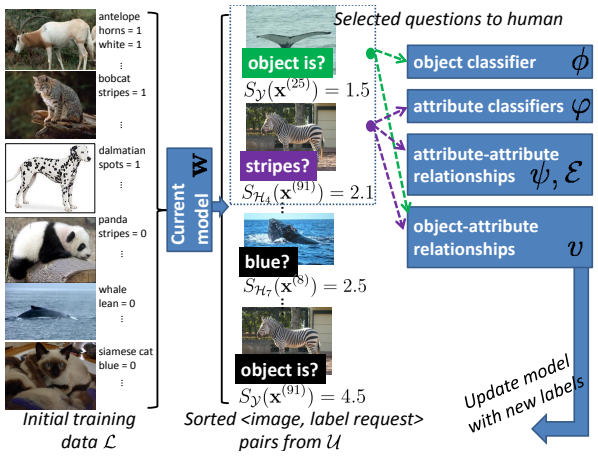

Figure 2. Overview of our approach. **Left:** the current model is determined by whatever labeled or partially labeled object and attribute data is available. Using that classifier, we score all ⟨image,label request⟩ pairs in the unlabeled pool according to their expected entropy reduction. **Center:** The N top scoring pairs are presented to an annotator with the targeted object or attribute question. **Right:** Depending on the answers and label types, the annotator responses influence different components of the full model, as signified by the two sets of dotted arrows. Note that the four rightmost boxes parallel the four terms of the main model in Eqn. 3. **Loop:** Finally, we loop back, and repeat the selection process using the newly strengthened model. Best viewed in color.

3.2.1 Annotation Set Definitions

Let $\mathcal{L}$ denote any labeled or partially labeled training data. Due to the different types of annotations and classifiers incorporated by the full model outlined above, we must maintain several separate training sets. As such, we think of $\mathcal{L}$ as containing several (potentially overlapping) sets: $\mathcal{L} = \{\mathcal{T}, \mathcal{T}_O, \mathcal{T}_{A_1}, \ldots, \mathcal{T}_{A_K}, \mathcal{T}_A\}$, where $\mathcal{T}$ contains fully labeled images used to train the full model $\mathbf{w}$, $\mathcal{T}_O$ contains object-labeled images used to train the object classifier that yields feature ϕ, each $\mathcal{T}_{A_m}$ contains attribute-labeled images used to train the attribute classifier that yields feature φ, and $\mathcal{T}_A$ contains attribute-labeled images used to compute the attribute relationship graph. Note that an annotation in $\mathcal{L} \setminus \mathcal{T}$ still affects the full model, because it alters the inner components on top of which $\mathbf{w}$ is learned.

Let $\mathcal{U}$ denote all unlabeled (or partially unlabeled) data. Similar to above, we maintain separate sets according to the label "state" of a given example: $\mathcal{U} = \{\mathcal{U}_O, \mathcal{U}_A\}$, where examples in $\mathcal{U}_O$ have no object label, and examples in $\mathcal{U}_A$ lack one or more attribute labels.

3.2.2 Entropy-Based Selection Function

At the onset, we are given some initial pool of labeled data in $\mathcal{L}$. At each iteration of active learning, we need to decide which image to have annotated and which annotation to request for it. Thus, we must rank the pool of candidate ⟨image,label request⟩ pairs in $\mathcal{U}$. A key point in our approach is that for a given image, there are $K+1$ options for the label query; it is either the object class, or one of the K attributes.

To this end, we define a selection function that scores the expected entropy reduction for a candidate request. Let $(y^{(i)}, h_1^{(i)}, \ldots, h_K^{(i)})$ denote the full labels for the i-th image $\mathbf{x}^{(i)}$. The total entropy over all labeled and unlabeled data given the labeled data $\mathcal{L}$ is defined as:

$$H(\mathcal{L}) = -\sum_{u=1}^{|\mathcal{L}\cup\mathcal{U}|}\sum_{l=1}^{|\mathcal{Y}|} P_{\mathcal{L}}\left(y^{(u)} = l|\mathbf{x}^{(u)}\right) \log P_{\mathcal{L}}\left(y^{(u)} = l|\mathbf{x}^{(u)}\right), \tag{4}$$

where $P_{\mathcal{L}}(y|\mathbf{x})$ denotes the posterior estimates obtained when the model is trained with labeled data $\mathcal{L}$. Note that entropy is measured over *object* predictions $\mathcal{Y}$, as that is the ultimate target label space.

In general, the unlabeled instance that maximizes the expected entropy reduction [22, 20] is:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{U}} \left(H(\mathcal{L}) - \sum_{l=1}^{|\mathcal{Y}|} P_{\mathcal{L}}\left(y = l|\mathbf{x}\right)\ \ H(\mathcal{L}\cup\langle\mathbf{x}, l\rangle)\right), \tag{5}$$

or equivalently, if we drop the constant current entropy value:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{U}} \left(\sum_{l=1}^{|\mathcal{Y}|} P_{\mathcal{L}}\left(y = l|\mathbf{x}\right)\ \ H(\mathcal{L}\cup\langle\mathbf{x}, l\rangle)\right). \tag{6}$$

In our case, we must consider expanding $\mathcal{L}$ by *either* the object label y or an attribute label h_m. Thus we define two intermediate entropy scoring terms:

$$S_{\mathcal{Y}}(\mathbf{x}^{(i)}) = \sum_{l=1}^{|\mathcal{Y}|} P_{\mathcal{L}}(y^{(i)} = l|\mathbf{x}^{(i)})\ \ H(\mathcal{L}\cup\langle\mathbf{x}^{(i)}, y^{(i)} = l\rangle), \tag{7}$$

where posteriors are obtained with the full object model, and

$$S_{\mathcal{H}_m}(\mathbf{x}^{(i)}) = \sum_{a=1}^{|\mathcal{A}|} P_{\mathcal{L}}(h_m^{(i)} = a|\mathbf{x}^{(i)})\ \ H(\mathcal{L}\cup\langle\mathbf{x}^{(i)}, h_m^{(i)} = a\rangle), \tag{8}$$

where posteriors are obtained from the model's inner attribute classifier φ. In both, $\mathcal{L}$ refers to the current labeled data. Note that $S_{\mathcal{Y}}$ and $S_{\mathcal{H}_m}$ are comparable in that they both reflect the entropy of the *object* label prediction.

Finally, the best image and label request is given by:

$$(\mathbf{x}^*, \boldsymbol{q}^*) = \arg\min_{\mathbf{x}\in\mathcal{U}, \boldsymbol{q}\in\{\mathcal{Y}, \mathcal{H}_1, \ldots, \mathcal{H}_K\}} S_{\boldsymbol{q}}(\mathbf{x}). \tag{9}$$

The lower the score in Eqn. 9, the more influence we expect the label request to have on the complete model. Because we consider the impact of a candidate labeling over all the

Algorithm 1 The proposed active learning approach.

Given: labeled data $\mathcal{L} = \{\mathcal{T}, \mathcal{T}_O, \mathcal{T}_{A_1}, \ldots, \mathcal{T}_{A_K}, \mathcal{T}_A\}$, and pool of unlabeled data $\mathcal{U} = \{\mathcal{U}_O, \mathcal{U}_A\}$.
Compute initial attribute relationship graph $(\mathcal{V}, \mathcal{E})$.
Compute features and train initial model using $\mathcal{L}$.
while Labeling effort still available **do**
  Compute $S_y(\mathbf{x})$ for all images in $\mathcal{U}_O$.
  Compute $S_{\mathcal{H}_m}(\mathbf{x})$ for all images in $\mathcal{U}_A$, for all yet-unlabeled attributes among $m = 1, \ldots, K$.
  Select the N most informative image-label pairs (Eqn. 9), and ask human annotator.
  Remove object-annotated images from $\mathcal{U}_O$.
  Remove fully attribute-annotated images from $\mathcal{U}_A$.
  Add new object-annotated images to $\mathcal{T}_O$.
  Add images with new labels for attribute m to $\mathcal{T}_{A_m}$.
  Infer values for any missing attribute labels for partially labeled images in $\mathcal{U}_A \cap \mathcal{T}_O$.
  Add those and fully attribute-labeled images to $\mathcal{T}_A$.
  Add images in $\mathcal{T}_O \cap \mathcal{T}_A$ to $\mathcal{T}$; remove them from U.
  Recompute inner classifiers' "features" using $\mathcal{L}$, update attribute graph.
  Retrain the full model $\mathbf{w}$ using $\mathcal{T}$.
end while

data and model components, this selection function reveals which attribute or object-based question is most valuable, achieving the intuition given in Fig. 1.

In order to compute the object class posterior probabilities required for entropy, we design a mapping from the raw $f_{\mathbf{w}}$ function outputs to multi-class probabilities. First we estimate the pairwise probabilities for any two object classes l_A, $l_B \in \mathcal{Y}$, by fitting a sigmoid to output values for $f_{\mathbf{w}}(\mathbf{x}, y = l_A) - f_{\mathbf{w}}(\mathbf{x}, y = l_B)$ on the training data in $\mathcal{T}$. The difference between output values mimics the form of the latent SVM label constraints. Then we use the pairwise coupling approach [30] to obtain multi-class probabilities from these pairwise probabilities. In this way, we essentially adapt Platt's method [16] to accommodate latent multi-class SVM outputs. For the attribute posteriors, we simply use Platt's method on the binary SVM scores.

Procedurally, computing the best request requires cycling over the unlabeled or partially labeled images. Then, for each label request we could make for the current image, we cycle over each possible label response, and (1) add it to the labeled set temporarily, (2) retrain the model, (3) evaluate entropy under the new model, and (4) weight the resulting entropy by the probability of the hypothesized label under the old model. To request more than one label per iteration, we simply take the N queries with the lowest $S_{\boldsymbol{q}}$ scores. Thus, one batch addition may include new labels for both objects and any of the attributes, and a given image may receive multiple labels.

3.2.3 Updates to Labeled and Partially Labeled Sets

Finally, we detail the implications that the above strategy has on the retraining step, whether adding true or hypothesized labels.

Recall that the model we are actively learning has two stages of training: the first updates the inner components (e.g., independent object or attribute classifiers), while the second updates the "outer" main parameters of $\mathbf{w}$ (see Eqn. 3). Updates to either of the two annotation types do not affect all inner components of the model at the same time, but they *do* always affect the full object prediction model parameters. In particular, new object labels are inserted into $\mathcal{T}_O$, and directly affect both the object classifier and learned object-attribute interaction terms. New attribute labels for the m-th attribute are inserted into $\mathcal{T}_{A_m}$, and directly affect the m-th attribute classifier, the attribute-attribute relationship graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and the object-attribute interaction terms. These dependencies are reflected by the dotted arrows in the example shown in Fig. 2.

Therefore, when we receive a new object label, we add it to $\mathcal{T}_O$ and remove it from $\mathcal{U}_O$. When we receive a new label for attribute m, we add it to $\mathcal{T}_{A_m}$; however, it is not removed from $\mathcal{U}_A$ until all other attributes for that image are obtained. If a new label happens to complete all labels for a given image (i.e., $\mathbf{x}^{(i)} \in \mathcal{T}_O \cap \mathcal{T}_A$), we remove it from $\mathcal{U}$ and insert it into $\mathcal{T}$.

In terms of updating the attribute relationship graph, if an object-labeled image in $\mathcal{T}_O$ has only partial attribute labels, there are two options: (1) a *conservative* approach, where we simply wait until all attribute labels are present before adding it to $\mathcal{T}_A$, or (2) a *partial* approach, where we add the image to $\mathcal{T}_A$ with its missing attribute labels inferred. To infer the missing labels, we add constraints in the linear programming problem that solves for h^* to reflect that any known attributes should be assigned their correct labels. After inferring these labels, we treat them as observed during training. For the partial approach, we keep the image in $\mathcal{U}_A$, so that its missing labels may still be added by a human annotator (if selected with active learning). We pursue this partial formulation in our experiments, as we expect more immediate impact of new labels to help the active learner.

Note that the partial update policy is applicable whether we are introducing a newly labeled instance received from an annotator (i.e., at the end of an active learning loop), or temporarily updating the model during the selection process. In the former they are permanent updates, while in the latter they are removed appropriately after the necessary posteriors are computed for Eqn. 9. Once we have updated the training and unlabeled sets accordingly, we retrain the inner classifiers, compute their features, and retrain the full model. See Alg. 1 for a recap of the method.

4. Experiments

We demonstrate our approach for object recognition on three challenging datasets. The main goal of our experiments is to demonstrate the advantage of choosing labels from both object and attribute types to validate the importance of a joint representation.

4.1. Datasets

We use the *Animals with Attributes* dataset introduced by Lampert et al. [15]. This dataset consists of 50 animal categories and 85 attributes. The attributes describe the fur color, fur patterns, size, anatomy, behavior, habitat etc. of the animals. We use three feature types as descriptors for these images: RGB histograms, PHOG, and rgSIFT, as given in [15]. We sample the categories and attributes from this dataset in different tasks, as described below.

We also use the *a-Yahoo-test* and *a-Pascal-train* datasets from [8]. The former consists of 12 classes and the latter of 20, which include animals, vehicles, household items, etc. These datasets use a set of 64 attributes, including shapes, textures, anatomy, and parts. Unlike the *Animals with Attributes* dataset, not every instance of a class in these datasets has the same attribute labels as all other instances in the same class. We use the provided bounding boxes to maintain the assumption that there is only one object per image. All datasets are fairly challenging because of appearance variation and have been used to evaluate several recent approaches for learning from attribute labels.

We show results for 4 different splits of classes, 2 from the *Animals with Attributes* dataset (*AwA-1* and *AwA-2*), 1 from *a-Yahoo*, and 1 from *a-Pascal*.[4] We use splits to manage the cost of the selection process. (Our method's cost grows linearly with the number of object classes and quadratically with the number of evaluated unlabeled images.) For each split, we sample 200 images from an unlabeled pool in each iteration and compute the quality score of each candidate ⟨image,label request⟩ pair in this set. The number of images in the full unlabeled pool and the separate test pool are: 1003/732 for *AwA-1*, 1002/993 for *AwA-2*, 903/287 for *a-Pascal*, and 703/200 for *a-Yahoo*. The initial number of labels on the learning curves (x-axis) is the number of training images per class[5] used to initialize the models times the number of categories times $(K + 1)$.

4.2. Baselines

In our experiments, we compare our full active method, **active-obj+attr (ours)**, which can request both object and attribute annotations, to a strong active baseline (**active-obj**) which is just like our full method but can only request object annotations during the active loop. We also compare these methods to a passive baseline (**random**), which is also competitive since it randomly requests labels from the pool of candidate object *and* attribute labels. To our knowledge, no existing active learning approach learns from both object and attribute labels, making these the two best baselines to compare. Note that **active-obj** has the disadvantage that additional object requests can only update the full model through the inner components, since an image has to be in both $\mathcal{T}_O$ and $\mathcal{T}_A$ to be added to $\mathcal{T}$. In one experiment, we also show the performance of **optimal selection**—this result reveals how entropy would be affected if we knew the true labels of the unlabeled images rather than relying on the expected entropy reduction value.

4.3. Results and Discussion

Active Learning Curves We first study the effect of our method and the baselines on the confidence of the correct label on a held-out test set. Fig. 3 reports the average probability of the correct label predicted by each method as a function of the number of labels added. These probabilities are computed over active learning iterations, where each time a selection of $N = 5$ labels is added for all methods. These standard learning curves aim to demonstrate that to achieve some given accuracy, our method usually requires fewer labels than any of the baselines. Higher values in the curve indicate more improvement in the classifier and a higher confidence of classification.

All three approaches improve upon the initial classifier with more labels, but at different rates per label. Our joint approach shows the most significant gains in accuracy with fewer labels. **Random** selection wastes annotator effort on less informative examples and labels. **Active-obj** is in general as good as or worse than **random**; the latter can happen since **random** has the advantage of additional attribute labels. The poorer performance of **active-obj** in comparison with **active-obj+attr (ours)** validates our main claim: it is more advantageous to select labels actively among both objects and attributes rather than just objects. Note that we show confidence rather than simply accuracy, since confidence reveals more fully how models have improved. For reference, the confidence results after training on all images in each class (excepting test images) are: .3810, .3745, .3852, .4020. Therefore, using only 4% of the total labels on average, we achieve 74.48% of the ultimate confidence level. In comparison, **random** achieves 67.39% and **active-obj** achieves 67.20% of the ultimate confidence level using the same number of labels (last point on the x-axis).

Actual Entropy Reduction Next, we examine how the uncertainty on the training and unlabeled sets changes as the different methods make their selections. Fig. 4 reports

[4] Splits *AwA-1* and *AwA-2* classes: *(hamster, hippopotamus, horse, humpback whale, killer whale)* and *(tiger, walrus, weasel, wolf, zebra)*. Split *a-Yahoo*: *(centaur, donkey, goat, monkey, wolf, zebra)*. Split *a-Pascal*: *(aeroplane, bicycle, boat, bus, car, motorbike, train)*.

[5] about 5; the exact numbers differ since we set the amount of initial training images proportionally to the number of images in each class

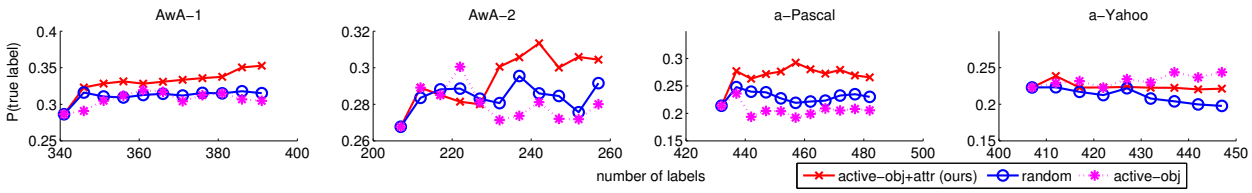

Figure 3. Representative learning curves from all three datasets showing the predicted probability of the correct label on a held-out test set with increasing number of labels obtained: first three are best, and fourth represents a failure case. Our approach is constantly more accurate than the baselines, indicating that jointly selecting from object and attribute labels is worthwhile. (Higher curves are better.)

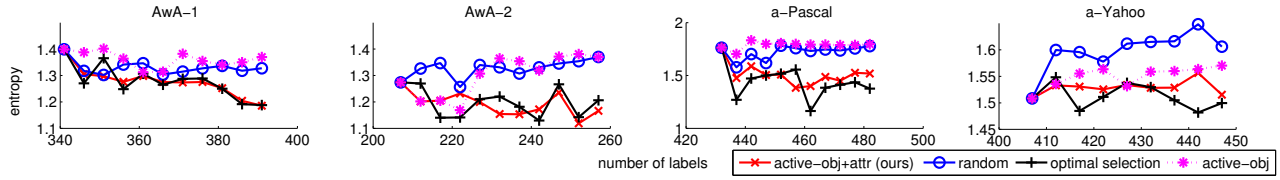

Figure 4. Entropy of all training and unlabeled examples with increasing number of obtained labels for our approach in comparison to optimal selection and the baselines (first three are good, and last is a failure case). As expected, our approach reduces the overall classification uncertainty faster than the baselines and similarly to optimal selection. (Lower curves are better.)

the mean entropy on $\mathcal{L} \cup \mathcal{U}$ as more labels are added for the three approaches and the optimal selection. We want entropy to decrease as more training data is added, so lower curves are better. Optimal selection shows the result of using ground truth information in order to compute the best possible selection based on entropy.[6] The overall entropy decreases steadily with more labels for both our approach and the optimal selection, showing that the classifier is able to better separate the examples into the different classes by jointly learning from object and attribute labels. Note that our approach performs quite similarly to optimal selection. The same is not true for the baselines, where the reduction in the overall uncertainty is slower.

The last figure in Fig. 4 shows a case where entropy is poorly estimated—compare our result versus the optimal selection in the fourth plot. This explains the failure case for the test set learning curves in the fourth plot in Fig. 3.

Qualitative Results To examine in more depth the selections made by our active learning approach, we present some qualitative results. In Fig. 5, we show the distribution of label requests for the object and each of the attribute labels. We see that the majority ($\sim 75\%$) of requests are for attribute labels. There is a slight tendency (not shown) that more object labels are requested earlier in the active learning loop. We see that the distribution for *a-Yahoo* is the least balanced one, which might indicate a particular relationship between objects and attributes that would explain the weaker performance of our method on this dataset. In Fig. 6, we show some sample requests that were made by our algorithm for *AwA-1*.

[6]This result would strictly be an upper bound to our approach, but since multiple labels are added at a time, the entropy reduction predicted for individual labels and the actual entropy reduction can differ.

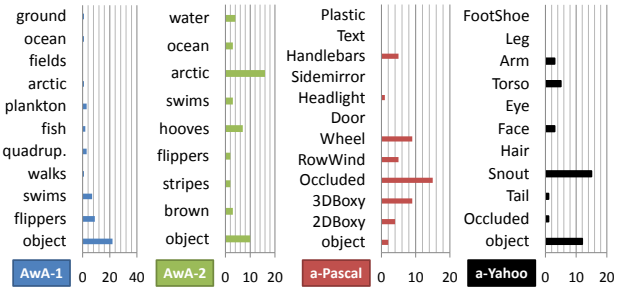

Figure 5. Distribution of requests per label type (object/attributes).

5. Conclusions

We propose a method for actively selecting the best object or attribute labels on images in a way that can simultaneously affect multiple object categories. Our results on three challenging datasets indicate that our method is indeed able to learn more quickly than either passive learning or a strong baseline approach that can only request object labels. The proposed strategy can be seen as a means to enhance multi-class object category learning, by efficiently strengthening models through shared attributes.

As future work, we would like to add a third request type which explicitly asks about the relationship between objects and attributes. We also plan to investigate alternative measures of uncertainty reduction and strategies for making our selection approach computationally scalable.

Acknowledgements

We thank Yang Wang for helpful discussions. This research was funded in part by grant ONR ATL N00014-11-1-0105, the Henry Luce Foundation, and grant NSF EIA-0303609.

Figure 6. Sample ⟨image,label⟩ requests that our method generates for *AwA-1*. The 1st request may be explained by the lack of dark brown hamsters in the training set. The 2nd and 3rd requests are due to the similarity to classes that have the attributes in question. The 4th and 5th requests show an image which confuses the system and appears in multiple labeling requests. The 6th image is likely ambiguous because it violates the assumption of one object per image.

References

[1] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010.

[2] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *ECCV*, 2010.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[4] T.-M.-T. Do and T. Artières. Large margin training for hidden Markov models and partially observed states. In *ICML*, 2009.

[5] G. Druck, B. Settles, and A. McCallum. Active learning by labeling features. In *EMNLP*, 2009.

[6] I. Endres, A. Farhadi, D. Hoiem, , and D. Forsyth. The benefits and challenges of collecting richer object annotations. In *ACVHL (in conjunction with CVPR)*, 2010.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, June 2010.

[8] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[9] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9), September 2010.

[10] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007.

[11] A. Gupta and L. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*, 2008.

[12] P. Jain and A. Kapoor. Active learning for large multi-class problems. In *CVPR*, 2009.

[13] A. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *CVPR*, 2009.

[14] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009.

[15] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[16] J. C. Platt. Probabilistic output for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 1999.

[17] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, and H.-J. Zhang. Two-dimensional active learning for image classification. In *CVPR*, 2008.

[18] H. Raghavan, O. Madani, and R. Jones. InterActive feature selection. In *IJCAI*, 2005.

[19] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where - and why? semantic relatedness for knowledge transfer. In *CVPR*, 2010.

[20] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, 2001.

[21] B. Russell, A. Torralba, K. Murphy, and W. Freeman. LabelMe: a database and web-based tool for image annotation. *IJCV*, 77:157–173, May 2008.

[22] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[23] B. Siddiquie and A. Gupta. Beyond active noun tagging: Modeling contextual interactions for multi-class active learning. In *CVPR*, 2010.

[24] A. Sorokin and D. Forsyth. Utility data annotation with Amazon Mechanical Turk. In *CVPR Workshop on Internet Vision*, 2008.

[25] S. Vijayanarasimhan and K. Grauman. Multi-level active prediction of useful image annotations for recognition. In *NIPS*, 2008.

[26] S. Vijayanarasimhan and K. Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *CVPR*, 2009.

[27] G. Wang and D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *ICCV*, 2009.

[28] Y. Wang and G. Mori. Max-margin hidden conditional random fields for human action recognition. In *CVPR*, 2009.

[29] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. In *ECCV*, 2010.

[30] T.-F. Wu, C.-J. Lin, and R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. In *Journal of Machine Learning Research*, volume 5, pages 975–1005, December 2004.

[31] C. Zhang and T. Chen. An active learning framework for content-based information retrieval. *IEEE Transactions on Multimedia*, 4(2), June 2002.