To appear, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2009.

What's It Going to Cost You?: Predicting Effort vs. Informativeness for Multi-Label Image Annotations

Sudheendra Vijayanarasimhan and Kristen Grauman
Department of Computer Sciences
University of Texas at Austin
{svnaras,grauman}@cs.utexas.edu

Abstract

Active learning strategies can be useful when manual labeling effort is scarce, as they select the most informative examples to be annotated first. However, for visual category learning, the active selection problem is particularly complex: a single image will typically contain multiple object labels, and an annotator could provide multiple types of annotation (e.g., class labels, bounding boxes, segmentations), any of which would incur a variable amount of manual effort. We present an active learning framework that predicts the tradeoff between the effort and information gain associated with a candidate image annotation, thereby ranking unlabeled and partially labeled images according to their expected "net worth" to an object recognition system. We develop a multi-label multiple-instance *approach that accommodates multi-object images and a mixture of strong and weak labels. Since the annotation cost can vary depending on an image's complexity, we show how to improve the active selection by directly predicting the time required to segment an unlabeled image. Given a small initial pool of labeled data, the proposed method actively improves the category models with minimal manual intervention.*

1. Introduction

Most visual recognition methods rely on labeled training examples where each class to be learned occurs prominently in the foreground, possibly with uncorrelated clutter surrounding it. In practice, the accuracy of a recognition algorithm is often strongly linked to the quantity and quality of the annotated training data available—having access to more examples per class means a category's variability can more easily be captured, and having richer annotations per image (e.g., a segmentation of object boundaries rather than a yes/no flag on object presence) means the learning stage need not infer which features are relevant to which object.

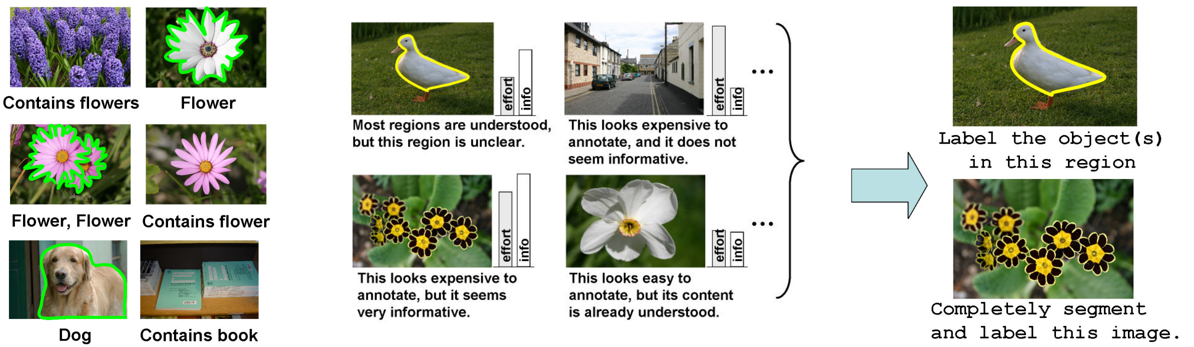
Unfortunately, this is a restrictive constraint, as substantial manual effort is needed to gather such datasets. Yet, not all images are equally informative. *Active learning* methods could potentially pinpoint a smaller set of uncertain examples for which labels should be requested [26, 8, 3, 14, 21], thereby reducing supervision without sacrificing much accuracy in the model.

However, in the general case, visual category learning does not fit the mold of traditional active learning approaches, which primarily aim to reduce the number of labeled examples required to learn a classifier, and almost always assume a binary decision task. When trying to choose informative image data to label for recognition, there are three important distinctions we ought to take into account.

First, most real-world images consist of multiple objects, and so should be associated with *multiple* labels simultaneously.[1] This means that an active learner must assess the value of an image containing some unknown combination of categories. Second, whereas in conventional learning tasks the annotation process consists of simply assigning a class label to an example, image annotation can be done at different levels—by assigning class labels, drawing a segmentation of object boundaries, or naming some region. This means an active learner must specify what *type* of annotation is currently most helpful, not just which example. Third, while previous methods implicitly assume that all annotations cost the same amount of effort (and thus minimize the total number of queries), the actual manual effort required to label images varies both according to the annotation type as well as the particular image example.

In order to handle these issues, we propose an active learning framework where the expected informativeness of any candidate image annotation is weighed against the predicted cost of obtaining it (see Figure 1). We devise a *multiple-instance, multi-label learning* (MIML) formulation that allows the system itself to choose which annotations to receive, based on the expected benefit to its current object models. After learning from a small initial set of la-

[1] Multi-label is thus more general than *multi-class*, where usually each example is assumed to represent an item from a single class.


Contains flowers
Flower
Flower, Flower
Contains flower
Dog
Contains book
effort
info
Most regions are understood, but this region is unclear.
This looks expensive to annotate, and it does not seem informative.
This looks expensive to annotate, but it seems very informative.
This looks easy to annotate, but its content is already understood.
...
Label the object(s) in this region
Completely segment and label this image.

(a) Labeled (and partially labeled) examples to build models

(b) Unlabeled and partially labeled examples to survey

(c) Actively chosen queries sent to annotators

Figure 1. Overview of the proposed approach. (a) We learn object categories from multi-label images, with a mixture of weak and strong labels. (b) The active selection function surveys unlabeled and partially labeled images, and for each candidate annotation, predicts the tradeoff between its informativeness versus the manual effort it would cost to obtain. (c) The most promising annotations are requested and used to update the current classifier.

beled images, our method surveys any available unlabeled data to choose the most promising annotation to receive next. After re-training, the process repeats, continually improving the models with minimal manual intervention.

Critically, our active learner chooses both which image example as well as what *type* of annotation to request: a complete image segmentation, a segmentation of a single object, or an image-level category label naming one of the objects within it. Furthermore, since any request can require a different amount of manual effort to fulfill, we explicitly balance the value of a new annotation against the time it might take to receive it. Even for the same type of annotation, some images are faster to annotate than others (e.g., a complicated scene versus an image with few objects). Humans (especially vision researchers) can easily glance at an image and roughly gauge the difficulty. But can we predict annotation costs directly from image features? Learning with data collected from anonymous users on the Web, we show that active selection gains actually improve when we account for the task's variable difficulty.

Our main contributions are a unified framework for predicting both the information content and the cost of different types of image annotations, and an active learning strategy designed for the MIML learning setting.

2. Related Work

A number of research threads aim at reducing the expense of obtaining well-annotated image datasets, from methods allowing weak supervision [23], to those that mine unlabeled images [18, 11]. Other techniques reduce training set sizes by transferring prior knowledge [5], or exploiting noisy images from the Web [6, 20]. Aside from such learning-based strategies, another approach is to encourage users to annotate images for free/fun/money [22, 15, 19].

Active learning for visual categories has thus far received relatively little attention. Active strategies typically try to minimize model entropy or risk, and have been shown to expedite learning for binary object recognition tasks [8], relevance feedback in video [26], dataset creation [3], and when there are correlations between image-level labels [14].

The multiple-instance learning (MIL) scenario has been explored for various image segmentation and classification tasks [12, 20, 28, 27]. Multi-label variants of MIL in particular are proposed in [28, 27], with impressive results. Active selection in the two-class MIL setting was recently explored in [16] and [21], where it is shown that a classifier learns faster if it can request both instance-level labels and bag-level labels. However, both previous active MIL methods are limited to learning from single-label examples and making binary decisions. In contrast, our approach makes it possible to actively learn multiple classes at once from images with multiple labels. This is an important distinction in practice, since images in a naturally occurring pool of unlabeled images will not be restricted to containing only one prominent object per image.

Overall, in contrast to this work, previous active learning methods for recognition only consider which examples to obtain a class label for to reduce uncertainty [26, 8, 3, 14], or else are limited to binary and/or single-label problems [8, 21]. None can learn from both multi-label image-level and region-level annotations. Finally, to our knowledge, no previous work has considered predicting the cost of an unseen annotation, nor allowing such predictions to strengthen active learning choices.

3. Approach

The goal of this work is to learn category models with minimum supervision under the real-world setting where each potential training image can be associated with multiple classes. Throughout, our assumption is that human effort is more scarce and expensive than machine cycles; thus our method prefers to invest in computing the best queries to make, rather than bother human annotators for an abundance of less useful labelings.

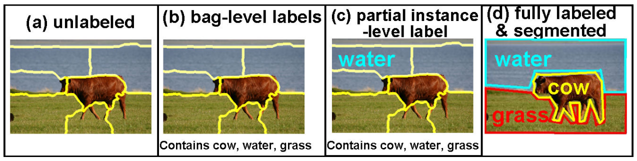

(a) unlabeled
(b) bag-level labels
(c) partial instance -level label
(d) fully labeled & segmented
Contains cow, water, grass
Contains cow, water, grass
water
water
cow
grass

Figure 2. In our MIML scenario, images are multi-label bags of regions (instances). Unlabeled images are oversegmented into regions (a). For an image with *bag-level* labels, we know which categories are present in it, but we do not know in which regions (b). For an image with some *instance-level* labels, we have labels on some of the segments (c). For a *fully annotated* image, we have true object boundaries and labels (d).

In the following, we introduce the MIML framework and define a discriminative kernel-based classifier that can deal with annotations at multiple levels (Section 3.1). Then, we develop a novel method to predict the cost of an annotation (Section 3.2.1). Finally, we derive a decision-theoretic function to select informative annotations in this multi-label setting, leveraging the estimated costs (Section 3.2.2).

3.1. Multi-label multiple-instance learning

An arbitrary unlabeled image is likely to contain multiple objects. At the same time, typically the easiest annotation to obtain is a list of objects present within an image. Both aspects can be accommodated in the multiple-instance multi-label learning setting, where one can provide labels at multiple levels of granularity (e.g., image-level or region-level), and the classifier learns to discriminate between multiple classes even when they occur within the same example.

In the following, we extend SVM-based MIL to the multi-label case. The main motivation of our design is to satisfy both the multi-label scenario as well as the needs of our active selection function. Specifically, we need classifiers that can rapidly be incrementally updated, and which produce probabilistic outputs to estimate how likely each label assignment is given the input.

In MIL [4], the learner is given *sets* (bags) of instances and told that at least one example from a positive bag is positive, while none of the members in a negative bag is positive. In the more general MIML setting, each instance within a bag can be associated with one of C possible class labels; therefore each bag is associated with multiple labels—whichever labels at least one of its instances has.

Formally, let $\{(X_1, L_1), (X_2, L_2), ...(X_N, L_N)\}$ denote a set of training bags and their associated labels. Each bag consists of a set of instances $X_i = \{x^i_1, x^i_2, ...x^i_{n_i}\}$, and a set of labels $L_i = \{l^i_1, l^i_2, \dots, l^i_{m_i}\}$, where n_i denotes the number of instances in X_i, and m_i denotes the number of labels in L_i. Note that often a bag has fewer unique labels than instances ($m_i \leq n_i$), since multiple instances may have the same label. Every instance x^i_j is associated with a description $\phi(x^i_j)$ in some kernel embedding space and some class label $l^i_k \in \mathbb{L} = \{1, \dots, C\}$, but with only the bag-level labels it is ambiguous which instance(s) belongs to which label. A bag X_i has label l if and only if it contains at least one instance with label l. Note that a labeled instance is a special case of a bag, where the bag contains only one example ($n_i = 1$), and there is no label ambiguity.

For our purposes, an image is a bag, and its instances are the oversegmented regions within it found automatically with a segmentation algorithm (see Figure 2). A bag's labels are tags naming the categories present within the image; a region (instance) label names the object in the particular region. Each region has a feature vector describing its appearance, color, shape, texture, etc. This follows the common use of MIL for images [12, 27, 21], but in the generalized multiple-instance multi-label case.

Our MIML solution has two components: first, we decompose the multi-class problem into a number of binary problems, in the spirit of standard one-vs-one classification; second, we devise a *Multi-label Set Kernel* that performs a weighting in kernel space to emphasize different instances within a bag depending on the category under consideration.

Each one-vs-one binary problem is handled by an SVM trained to separate bags containing label l_i from those containing l_j, for all i, j. For the single-label case, one can average a bag's features to make a single feature vector summarizing all its instances: $\phi(X_i) = \frac{1}{|X_i|} \sum_{j=1}^{n_i} \phi(x^i_j)$, and then train an SVM with instances and bags; this is the Normalized Set Kernel (NSK) approach of [7]. However, in the multi-label case, some bags could be associated with *both* labels l_i and l_j. Simply treating the image as a positive example when training both classes would be contradictory. Intuitively, when training a classifier for class l_i, we want a bag to be represented by its component instances that are most likely to have the label l_i, and to ignore the features of its remaining instances. Of course, with bag-level labels only, the assignment of labels to instances is unknown.

We therefore propose a Multi-label Set Kernel that weights the feature vectors of each instance within the bag according to the estimated probability that the instance belongs to the class. That way if an instance has a high chance of belonging to the given class, then its feature vector will dominate the representation. To this end, we design a class-specific feature representation of bags. Let $X = \{x_1, \dots, x_n\}$ be a bag containing labels $L = l_1, \dots, l_m$ (where here we drop the example index i for brevity). We define the class-specific feature vector of X for class l_k as

$$\phi\left(X^{(l_k)}\right) = \sum_{j=1}^{n} \Pr(l_k|x_j)\phi(x_j), \qquad (1)$$

which weights the component instances by their probability of being associated with the class label under consideration. Here $\Pr(l_k|x_j)$ denotes the *true* probability that instance x_j belongs to category l_k, which we approximate as $\Pr(l_k|x_j) \approx p(l_k|x_j)$, where $p(l_k|x_j)$ is the posterior probability output by the classifier using the training data seen thus far. For a single instance (or equivalently, a single-instance bag), there is no label ambiguity, so the instance is simply represented by its feature vector.

For generic kernels, we may not know the feature space mapping $\phi(x)$ needed to explicitly compute Eqn (1). Instead, we can apply the same feature weights via the kernel value computation. Let X_1 and X_2 be bags associated with labels l_1 and l_2, respectively, that are currently being used to construct a classifier separating classes l_1 and l_2. Then the kernel value between bags X_1, X_2 is given by

$$\mathcal{K}(X_1^{(l_1)}, X_2^{(l_2)}) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} p(l_1|x_i^1)\, p(l_2|x_j^2)\, \mathcal{K}(x_i^1, x_j^2),$$

where $\mathcal{K}(x_i^1, x_j^2) = \phi(x_i^1)^T \phi(x_j^2)$ is the kernel value computed for instances x_i^1 and x_j^2, and $p(l_1|x_i^1), p(l_2|x_j^2)$ are the posteriors from the current classifiers. Note that because the kernel is parameterized by the label under consideration, a multi-label bag can contribute multiple different $\langle$feature,label$\rangle$ pairs to the training sets of a number of the one-vs.-one classifiers.

Our Multi-label Set Kernel can be seen as a generalization of the NSK [7], which is restricted to single-label binary classification. It is also related to the kernel in [10], where weights are set using a Diverse Density function.

The proposed kernel is valid for both instances and bags, and thus can be used to build SVMs for all required component binary problems. Each SVM can accept novel instances or bags: the feature for an input instance is unchanged, while an input bag is weighted according to Eqn (1). Given a new input X_{new}, we (a) run it through all $\frac{1}{2}C \times (C-1)$ classifiers, (b) compute the $\frac{1}{2}C \times (C-1)$ resulting two-class posteriors using [13], and, finally, (c) map those posteriors to the multi-class posterior probabilities $p(l|X_{new})$ for each label $l \in \{1, \ldots, C\}$. For this last step we use the pairwise coupling approach of [25].

3.2. Active multi-level selection of multi-label annotations

Thus far we have defined the multi-label learner, the basic classifier with which we want to actively learn. Next we describe our strategy to do active selection among candidate annotations. For each candidate, the selection function measures its expected informativeness and subtracts its predicted cost. We first address how to predict cost (Section 3.2.1), followed by informativeness (Section 3.2.2).

3.2.1 Predicting the cost of an annotation

There are three possible types of annotation request: the classifier can ask for a label on a bag, a label on an instance within a bag, or a label on all instances within a bag. A label on a bag serves as a "flag" for class membership, which is ambiguous because we do not know which of the instances in the bag are associated with the label. A label on an instance unambiguously names the class in a single image region, while labeling all instances within a bag corresponds to fully segmenting and labeling an image (see Figure 2).

Traditional active learning methods assume equal manual effort per label, and thus try to minimize the total number of queries made to the annotator. In reality annotation costs will vary substantially from image to image, and from type to type. Thus, the standard "flat cost" implied by traditional active learners is inadequate. To best reduce human involvement, the active learner needs a quantitative measure of the effort required to obtain any given annotation.

The goal is to accurately predict annotation time based on image content alone—that is, without actually obtaining the annotation, we need to estimate how long it will take a typical annotator to complete it. It seems plausible that the difficulty level could be predicted based on the image's features. For an extreme example, if an image contains a single color it most likely contains only one object, and so it should not be difficult to segment. If the image has significant responses to a large number of filters, then it may be highly cluttered, and so it could take a long time.

Thus, we propose to use supervised learning to estimate the difficulty of segmenting an image. It is unclear what features will optimally reflect annotation difficulty, and admittedly high-level recognition itself plays some role. We select candidate low-level features, and then use multiple kernel learning [1] to select those most useful for the task. We begin with some generic features that may be decent indicators of image complexity: a histogram of oriented gradients, a hierarchical grid where each cell measures edge density, a color histogram, and a grayscale histogram.

We gather the data online, using Amazon's Mechanical Turk system, where we can pay anonymous users to segment images of our choosing. The users are given a polygon-drawing tool to superimpose object boundaries, and are instructed to name and outline every major object. The system times their responses. Thus the labels on the training images will be the times that annotators needed to complete a full annotation. To account for noise in the data collection, we collect a large number of user responses per image. Even if users generally have the same relative speeds (faster on easy ones, slower on harder ones), their absolute speeds may vary. Therefore, to make the values comparable, we normalize each user's times by his/her mean and use the average time taken on an image to be its target label.

Given the image features and time labels, we train an SVM (not to be confused with the MIML classifier above) that, given an image, can predict the amount of manual effort it takes to annotate it. We can construct either coarse classifiers that categorize images into a discrete range of difficulty levels, or else regressors that can provide more precise time estimates. We pursue both in Section 4.

From this we can build a cost function $\mathcal{C}(\mathbf{z})$ that takes a candidate annotation $\mathbf{z}$ as input, and returns the predicted time requirement (in seconds) as output. When $\mathbf{z}$ is a candidate full segmentation, we apply the learned function to the image. When $\mathbf{z}$ is a request for a tag (bag-level label), we set $\mathcal{C}(\mathbf{z})$ as the cost estimated using similar time-based experiments. Finally, when $\mathbf{z}$ entails outlining a single object, we estimate the cost as the full image's predicted time, divided by the number of segments in the image.

3.2.2 Predicting the informativeness of an annotation

Given this learned cost function, we can now define the complete MIML active learning criterion. Inspired by the classic notion of the *value of information* (VOI), and by previous binary single-label active learners [9, 21], we derive a measure to gauge the relative risk reduction a new multi-label annotation may provide. The main idea is to evaluate the candidate images and annotation types, and predict which combination (of image+type) will lead to the greatest net decrease in risk for the current classifier, when each choice is penalized according to its expected manual effort.

Defining the risk terms. At any stage in the learning process the dataset can be divided into three different pools: $\mathcal{X}_U$, the set of unlabeled examples (bags and instances); $\mathcal{X}_L$, the set of labeled examples; and $\mathcal{X}_P$, the set of partially labeled examples, which contains all bags for which we have only a partial set of bag-level labels. Let r_l denote the risk associated with misclassifying an example belonging to class l. The risk associated with $\mathcal{X}_L$ is:

$$\mathcal{R}(\mathcal{X}_L) = \sum_{X_i \in \mathcal{X}_L} \sum_{l \in L_i} r_l \left(1 - p(l|X_i)\right), \qquad (2)$$

where $p(l|X_i)$ is the probability that X_i is classified with label l. Here, X_i is again used to denote both instances and bags and L_i its label(s). If X_i is a training instance it has only one label, and we can compute $p(l|X_i)$ via the current MIML classifier.

If X_i is a multi-label bag in the training set, we can compute the probability it receives label l as follows:

$$p(l|X_i) = p\left(l|x_1^i, \ldots, x_{n_i}^i\right) = 1 - \prod_{j=1}^{n_i}(1 - p(l|x_j^i)). \quad (3)$$

For a bag to *not* belong to a class, it must be the case that none of its instances belong to the class. Thus the probability of a bag *not* having a label is equivalent to the probability that *none* of its instances have that class label.

The corresponding risk for the unlabeled data is then:

$$\mathcal{R}(\mathcal{X}_U) = \sum_{X_i \in \mathcal{X}_U} \sum_{l=1}^{C} r_l (1 - p(l|X_i)) \Pr(l|X_i), \qquad (4)$$

where we compute the probabilities for bags using Eqn. 3, and $\Pr(l|X_i)$ is the true probability that unlabeled example X_i has label l, approximated as $\Pr(l|X_i) \approx p(l|X_i)$.

For the partially labeled data, the risk is:

$$\begin{aligned}\mathcal{R}(\mathcal{X}_P) &= \sum_{X_i \in \mathcal{X}_P} \sum_{l \in L_i} r_l \left(1 - p(l|X_i)\right) \qquad (5)\\ &+ \sum_{l \in U_i} r_l \left(1 - p(l|X_i)\right) p(l|X_i),\end{aligned}$$

where $U_i = \mathbb{L} \setminus L_i$.

The risk parameter r_l should be set to reflect the damage done by a single misclassification, using the same units as the cost function in Section 3.2.1. Intuitively, it corresponds to the amount of user time that might be wasted by mislabeling an example.

Computing the value of information. The total cost $T(\mathcal{X}_L, \mathcal{X}_U, \mathcal{X}_P)$ associated with a given snapshot of the data is the total misclassification risk, plus the cost of obtaining all the labeled data thus far:

$$T(\mathcal{X}_L, \mathcal{X}_U, \mathcal{X}_P) = \mathcal{R}(\mathcal{X}_L) + \mathcal{R}(\mathcal{X}_U) + \mathcal{R}(\mathcal{X}_P) + \sum_{X_i \in \mathcal{X}_B} \sum_{l \in L_i} \mathcal{C}(X_i^l),$$

where $\mathcal{X}_B = \mathcal{X}_L \cup \mathcal{X}_P$, and $\mathcal{C}(\cdot)$ is defined as above.

We measure the utility of obtaining a particular annotation by predicting the change in total cost that would result from the addition of the annotation to $\mathcal{X}_L$. Therefore, the value of information for an annotation $\mathbf{z}$ is:

$$\begin{aligned}VOI(\mathbf{z}) &= T\left(\mathcal{X}_L, \mathcal{X}_U, \mathcal{X}_P\right) - T\left(\hat{\mathcal{X}}_L, \hat{\mathcal{X}}_U, \hat{\mathcal{X}}_P\right) \qquad (6)\\ &= \mathcal{R}(\mathcal{X}_L) + \mathcal{R}(\mathcal{X}_U) + \mathcal{R}(\mathcal{X}_P)\\ &\quad - \left(\mathcal{R}(\hat{\mathcal{X}}_L) + \mathcal{R}(\hat{\mathcal{X}}_U) + \mathcal{R}(\hat{\mathcal{X}}_P)\right) - \mathcal{C}(\mathbf{z}),\end{aligned}$$

where $\hat{\mathcal{X}}_L, \hat{\mathcal{X}}_U, \hat{\mathcal{X}}_P$ denote the set of labeled, unlabeled and partially labeled data after obtaining annotation $\mathbf{z}$. If $\mathbf{z}$ is a complete annotation, then $\hat{\mathcal{X}}_L = \mathcal{X}_L \cup \mathbf{z}$; otherwise, $\hat{\mathcal{X}}_P = \mathcal{X}_P \cup \mathbf{z}$, and the example associated with $\mathbf{z}$ is removed from $\mathcal{X}_U$ and $\mathcal{X}_P$ as appropriate.

A high VOI for a given input denotes that the total cost would be decreased by adding its annotation. So, the classifier seeks annotations that give maximal VOI values.

Estimating risk for candidate annotations. The VOI function relies on estimates for the risk of yet-unlabeled data, so we must predict how the classifier will change given the candidate annotation, without actually knowing its label(s). We estimate the total risk induced by incorporating a candidate annotation $\mathbf{z}$ using the expected value: $\mathcal{R}(\hat{\mathcal{X}}_L) + \mathcal{R}(\hat{\mathcal{X}}_U) + \mathcal{R}(\hat{\mathcal{X}}_P) \approx E[\mathcal{R}(\hat{\mathcal{X}}_L) + \mathcal{R}(\hat{\mathcal{X}}_U) + \mathcal{R}(\hat{\mathcal{X}}_P)]$.

This expected value is straightforward to compute for a candidate instance x_z; we simply remove the unlabeled example from $\mathcal{X}_U$, temporarily add it to $\mathcal{X}_L$ with each of the possible C labels (in turn), evaluate the risk using the updated classifier, and weight each term by $p(l|x_z)$. We use incremental updates to make this fast [2]. Similarly, if the candidate annotation would add an image-level label to an unlabeled bag X_z, we do the same, computing the probabilities using Eqn. 3. The more complex case is for candidate

Approach	Ave. AUROC (img)	Ave. AUROC (region)
Ours	0.896 ± 0.00	0.91 ± 0.01
MLMIL [27]	0.902	0.863

Table 1. Five-fold cross-validation accuracies when training with only image-level labels.

full segmentations. Each one entails C^M possible label assignments, making a direct computation of the expectation impractical. For these, we estimate the expected total risk with Gibbs sampling. We presented a related sampling procedure in [21] for two-class single-label data.

3.3. Summary of the algorithm

We can now actively select multi-label, multi-level image annotations so as to maximize the expected benefit relative to the manual effort expended. The MIML classifier is initially trained using a small number of tagged images. To get each subsequent annotation, the active learner surveys all remaining unlabeled and partially labeled examples, computes their VOI, and requests the label for the example with the maximal value. After the classifier is updated with this label, the process repeats. The final classifier can predict image- and region-level labels, in binary or multi-class settings.

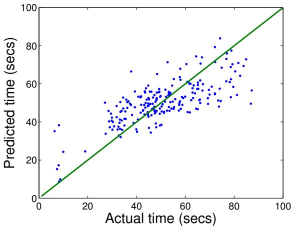
4. Results

To validate our method we use the publicly available MSRC dataset, since it contains multi-label images and a variable number of objects per image, and also allows comparisons with another MIML approach and other state-of-the-art methods. The MSRC v2 contains 591 images and 21 classes, with 240 images and 14 classes in the (subset) v1. In all experiments we use an RBF kernel with $\gamma = 10$, and set the SVM parameters (including the sigmoid parameters for [13]) based on cross-validation. We ignore all "void" regions. We evaluate three aspects of our approach: (1) its accuracy when learning from multi-label examples, (2) its ability to accurately predict annotation costs, and (3) its effectiveness as an active learner to reduce manual effort.

Multi-label learning. We divide the MSRC v2 into 5 folds containing about an equal number of images, as in [27]. We choose one part as the test set, one to set parameters, and train on the rest. We segment the images with Normalized Cuts, and obtain texton and color histograms for each blob, as in [17]. Each image is a bag, and each segment is an instance. To learn the MIML classifier, we use only image-level (bag-level) labels, i.e., we withhold all the pixel-level labels during classifier training.

Table 1 shows the average AUROC when predicting labels on new *images* (2^{nd} column) or new *regions* (3^{rd} column). For image-level prediction our results are comparable to the state-of-the-art in MIML [27], whereas for region-level prediction we achieve a notable improvement (0.91 vs. 0.86). This appears to be a direct consequence of our Multi-label Set Kernel, which weighs the region descriptors so as to represent an image by its most relevant instances for each image-level label. As a result, we are able to directly separate novel regions from each class within a new image, and not just name objects that occur in it.

User	Number of images	Accuracy (%)
User 1	160	68.75
User 2	188	72.34
User 3	179	70.95
User 4	151	72.85
User 5	167	59.88
User 6	164	63.41
User 7	169	67.46
User 8	179	79.33
All users	210	**73.81**



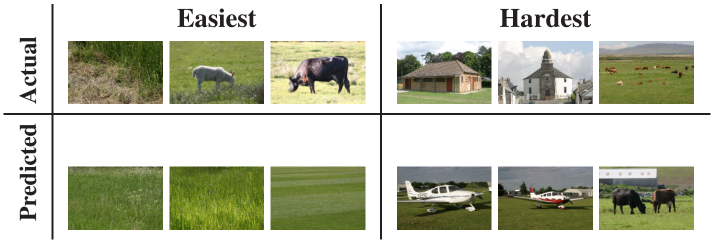


Figure 3. **Top Left:** Accuracy of our cost function in predicting "easy" vs. "hard", both for user-specific and user-independent classifiers. **Top Right:** Scatter-plot of the actual time taken by users to segment an image vs. the value predicted by our cost function, for the 240 images in the MSRC v1. **Images:** The easiest and hardest images to annotate based on actual users' timing data (top), and the predictions of our cost function on novel images (bottom).

Next we compare against the approaches of [17] and [24], which use pixel-level labels (full segmentations) to train a multi-class classifier. Restricting our method to only image-level labels, we obtain a region-based accuracy of $64.1\% \pm 2.9$ over 5 trials of approximately equal train-test splits. In comparison, the accuracy obtained for the same test scenario is 70.5% in [17], and 67.6% in [24]. Thus with much less manual training effort (image tags), our method performs quite competitively with methods trained with full segmentations; this illustrates the advantage of the multi-label multi-instance learner. Using the NSK [7], which essentially removes our kernel weight mapping, the accuracy for this test would only be $55.95\% \pm 1.43$.

Annotation cost prediction. To train our cost function, we gather data with Amazon's Mechanical Turk. Users are required to completely segment images from the 14-class MSRC v1 dataset while a script records the time taken per image. We collected 25-50 annotations per image from different users. Users could skip images they preferred not to segment; the fact that most users skipped certain images (Figure 3) supports our hypothesis that segmentation difficulty can be gauged by glancing at the image content.

We train both classifiers that can predict "easy" vs. "hard", and regressors that can predict the actual time in

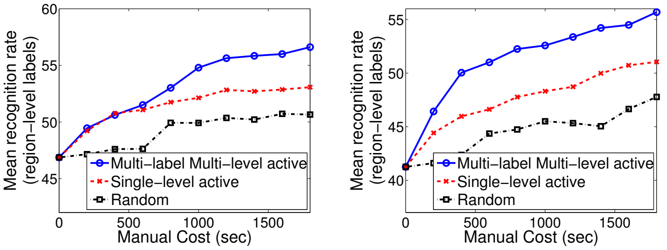


Figure 4. Learning curves when actively or randomly selecting multi-level and single-level annotations. **Left:** Region-level accuracy for the 21-class MSRC v2 dataset plotted against ground truth cost. **Right:** Region-level accuracy when 80 random images were added to the unlabeled pool.

seconds. To divide the training set into easy and hard examples, we simply use a threshold at the mean time taken on all images. Using the feature pool described in Section 3.2.1, we perform multiple-kernel learning to select feature types for both the user-specific data and the combined datasets. The edge density measure and color histograms received the largest weights, with the rest near zero. Figure 3 (left) shows the leave-one-out cross validation (loo-cv) result when classifying images as easy or hard, for the users for whom we had the most data. For the majority, accuracy is well above chance. Most of the errors may largely be due to our arbitrary division between what is easy or hard based on the mean.

To train a regressor we use the raw timing data and the same set of features. Figure 3 shows examples that were easiest and hardest to segment, as measured by the ground truth actual time taken for at least 8 users. Alongside, we show the examples that our regressor predicts to be easiest and hardest (from a separate partition of the data). These examples are intuitive, as one can imagine needing a lot more clicks to draw polygons on the many objects in the “hardest” set. Figure 3 also plots the actual time taken by users on an image against the value predicted by our cost function, as obtained with loo-cv for all 240 images in the MSRC v1 dataset. The rms difference between the actual and predicted times is 11.1 s, with an average prediction error of 22%. In comparison, predicting a constant value of 50 s (the mean of the data) yields an average prediction error of 46%. Given that the actual times vary from 8 to 100 s, and that the average cross-annotator disagreement was 18 s, an average error of 11 s seems quite good.

Active selection from MIML data. Next we demonstrate the impact of using our multi-label active selection function to choose from different types of annotations.

We construct the initial training set such that each class appears in at least 5 images, and use image-level labels. The rest of the training set forms the unlabeled pool of data. The active learner can request either complete segmentations or region-level labels from among the initial training examples, or image-level labels from any unlabeled example. We set $r_l = 50$ for all classes, which means that each misclassification is worth 50 s of user time.[2] For this experiment we fix the costs per type using the mean times from real users: 50 s for complete segmentations, 10 s for a region outline, and 3 s for a flag. We compare our approach to a “passive” selection strategy, which uses the same classifier but picks labels to receive at random, as well as a single-level active baseline (traditional active learning) that uses our VOI function, but only selects from unlabeled regions. All methods are given a fixed cost and allowed to make a sequence of label requests (to an oracle) until the cost is used up.

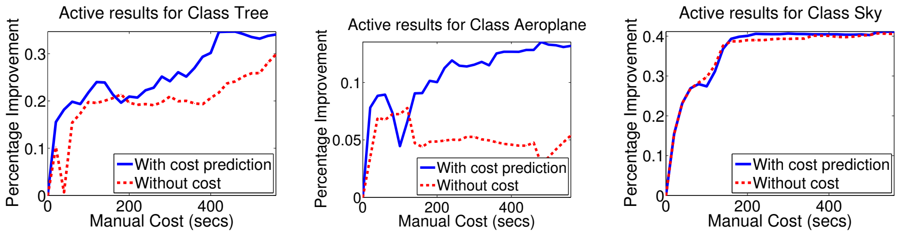
Figure 4 shows the resulting learning curves for the MSRC v2. Accuracy is measured as the average value of the diagonal of the confusion matrix for region-level predictions on the test set. All results are averaged over 5 random trials. The proposed multi-level active selection yields the steepest learning curves. Random selection lags behind, wasting annotation effort on less informative examples. Single-level active is preferable to random selection, yet we get best results when our active learner can choose between multiple types of annotations, including segmentations or image flags. The total gains after 1800s are significant, given the complexity of the 21-way classification problem with a test set containing 1129 image regions. Note that the random selection curve is probably an over-estimate of its quality; since we limit the unlabeled pool to only images from the MSRC, any example it requests is going to be fairly informative. Figure 4 (right) shows results for the same setting when 80 random images are added to the unlabeled pool, indicating that when uninformative images are present random selection lags even further behind.

Active selection with a learned cost function. Finally, we show the impact of using the predicted cost while making active choices. We train a binary multi-instance classifier for each category using image labels on $\frac{4}{5}$-th of the data per class, in 5 different runs. The rest is used for testing. We compare two MIL active learners: one using cost prediction, and one assigning a flat cost to annotations. At test time, both learners are “charged” the ground truth cost of getting the requested annotation.

Figure 5 shows representative (good and bad) learning curves, with accuracy measured by the AUROC value. For Tree and Airplane, using the predicted cost leads to much better accuracies at a lower cost, whereas for Sky there is little difference. This may be because most ‘sky’ regions look similar and take similar amounts of time to annotate.

Figure 5 (right) shows the cost required to improve the base classifier to different levels of accuracy. The 4^{th} column shows the relative time savings our cost prediction enables over a cost-blind active learner that uses the same

[2] The parameter r_l should reflect the real cost of a classification mistake. We set it to 50 since an error made by the automatic labeling would take around 50 s to manually fix for the average image.



% imp.	Cost(secs)		% Cost saved
	CP	NC	
5	11.40	11.52	+1.07
10	24.52	31.41	+21.94
15	45.25	63.24	+28.45
20	165.85	251.10	+33.95
25	365.73	543.69	+32.73

Figure 5. Representative learning curves (left) and a summary over all 14 classes (right) when using active selection with the learned cost predictor, as compared to a baseline that makes active selections using a flat cost value. **Rightmost:** Savings in cost when using cost prediction within the active learner. **CP** refers to using cost prediction and **NC** is without cost. Overall, our active selection takes less effort to attain the same level of accuracy as a cost-blind active learner.

selection strategy. For larger improvements, predicting the cost leads to noticeably greater savings in manual effort—over 30% savings to attain a 25% accuracy improvement.[3]

With our implementation of the incremental SVM technique of [2] it takes on average 0.5 secs to evaluate a single region and 20 secs to evaluate a bag (image) on a 1.6 GHz PC. We are currently considering ways to alleviate the computational cost. However, even without real-time performance, a distributed framework for image labeling that involves multiple annotators could be run efficiently.

5. Conclusions

We proposed an active learning framework that not only chooses examples based on their information content, but also on the predicted cost of obtaining the information. Our framework operates in the challenging real-world domain of multi-label images, with multiple possible types of annotations. Our results demonstrate that (1) the active learner obtains accurate models with much less manual effort than typical passive learners, (2) we can fairly reliably estimate how much a putative annotation will cost given the image content alone, and (3) our multi-label, multi-level strategy outperforms conventional active methods that are restricted to requesting a single type of annotation.

Acknowledgements. Many thanks to Alex Sorokin for helping us arrange the Mechanical Turk data collection. This research was supported in part by NSF CAREER 0747356, Microsoft Research, Texas Higher Education Coordinating Board award # 003658-01-40-2007, DARPA VIRAT, NSF EIA-0303609, and the Henry Luce Foundation.

References

[1] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Fast Kernel Learning using Sequential Minimal Optimization. Technical Report UCB/CSD-04-1307, Feb 2004.

[2] G. Cauwenberghs and T. Poggio. Incremental and Decremental Support Vector Machine Learning. In *NIPS*, 2000.

[3] B. Collins, J. Deng, K. Li, and L. Fei-Fei. Towards Scalable Dataset Construction: An Active Learning Approach. In *ECCV*, 2008.

[4] T. Dietterich, R. Lathrop, and T. Lozano-Perez. Solving the Multiple Instance Problem with Axis-Parallel Rectangles. *Artificial Intelligence*, 1997.

[5] L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories. In *ICCV*, '03.

[6] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning Object Categories from Google's Image Search. In *ICCV*, 2005.

[7] T. Gartner, P. Flach, A. Kowalczyk, and A. Smola. Multi-Instance Kernels. In *ICML*, 2002.

[8] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active Learning with Gaussian Processes for Object Categorization. In *ICCV*, 2007.

[9] A. Kapoor, E. Horvitz, and S. Basu. Selective Supervision: Guiding Supervised Learning with Decision-Theoretic Active Learning. In *IJCAI*, 2007.

[10] J. T. Kwok and P. Cheung. Marginalized Multi-Instance Kernels. In *In IJCAI*, 2007.

[11] Y. Lee and K. Grauman. Foreground Focus: Finding Meaningful Features in Unlabeled Images. In *BMVC*, 2008.

[12] O. Maron and A. L. Ratan. Multiple-Instance Learning for Natural Scene Classification. In *In ICML*, 1998.

[13] J. Platt. *Advances in Large Margin Classifiers*, chapter Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. MIT Press, 1999.

[14] G. Qi, X. Hua, Y. Rui, J. Tang, and H. Zhang. Two-Dimensional Active Learning for Image Classification. In *CVPR*, 2008.

[15] B. Russell, A. Torralba, K. Murphy, and W. Freeman. Labelme: a Database and Web-Based Tool for Image Annotation. Technical report, MIT, 2005.

[16] B. Settles, M. Craven, and S. Ray. Multiple-Instance Active Learning. In *NIPS*, 2008.

[17] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation. In *In ECCV*, 2006.

[18] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering Object Categories in Image Collections. In *ICCV*, '05.

[19] A. Sorokin and D. Forsyth. Utility Data Annotation with Amazon Mechanical Turk. In *CVPR Workshops*, 2008.

[20] S. Vijayanarasimhan and K. Grauman. Keywords to Visual Categories: Multiple-Instance Learning for Weakly Supervised Object Categorization. In *CVPR*, 2008.

[21] S. Vijayanarasimhan and K. Grauman. Multi-Level Active Prediction of Useful Image Annotations for Recognition. In *NIPS*, 2008.

[22] L. von Ahn and L. Dabbish. Labeling Images with a Computer Game. In *CHI*, 2004.

[23] M. Weber, M. Welling, and P. Perona. Unsupervised Learning of Models for Recognition. In *ECCV*, 2000.

[24] J. Winn, A. Criminisi, and T. Minka. Object Categorization by Learned Universal Visual Dictionary. In *ICCV '05*, 2005.

[25] T.-F. Wu, C.-J. Lin, and R. C. Weng. Probability Estimates for Multi-Class Classification by Pairwise Coupling. *JMLR*, 2004.

[26] R. Yan, J. Yang, and A. Hauptmann. Automatically Labeling Video Data using Multi-Class Active Learning. In *ICCV*, 2003.

[27] Z.-J. Zha, X.-S. Hua, T. Mei, J. Wang, G.-J. Qi, and Z. Wang. Joint Multi-Label Multi-Instance Learning for Image Classification. In *CVPR*, 2008.

[28] Z. H. Zhou and M. L. Zhang. Multi-Instance Multi-Label Learning with Application to Scene Classification. In *NIPS*, 2006.

[3] Note that since we have only 240 ground truth images with 50 users' annotation times recorded each, the total savings we can illustrate in this experiment in absolute terms is necessarily somewhat limited.