Cost-Sensitive Active Visual Category Learning

Sudheendra Vijayanarasimhan · Kristen Grauman

Abstract 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 realistic images containing multiple objects and allows the category-learner to strategically choose what annotations it receives from 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. Our approach accounts for the fact that the optimal use of manual effort may call for a combination of labels at multiple levels of granularity, as well as accurate prediction of manual effort. As a result, it is possible to learn more accurate category models with a lower total expenditure of annotation effort. Given a small initial pool of labeled data, the proposed method actively improves the category models with minimal manual intervention.

Keywords Visual category learning · Active learning · Multi-label · Multiple-instance learning · Cost prediction · Cost sensitive learning

1 Introduction

One of the primary challenges in computer vision research is the problem of recognizing generic object categories. It is challenging on a number of levels: objects of the same class may exhibit an incredible variability in appearance, real-world images naturally contain large amounts of irrelevant background "clutter", and subtle context cues can in many cases be crucial to proper perception of objects. Nonetheless, recent advances have shown the feasibility of learning accurate models for a number of well-defined object categories.

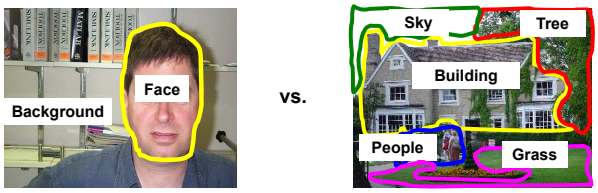
Most visual recognition methods rely on labeled training examples where each class to be learned occurs prominently in the foreground, possibly with uncorrelated clutter surrounding it. In practice, the accuracy of a recognition algorithm is often strongly linked to the quantity and quality of the annotated 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, suggesting that a wiser and more targeted use of human attention could make the visual category learning process more effective.

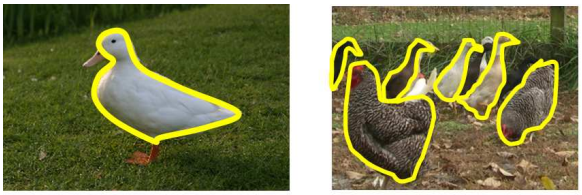
Active learning strategies provide a way to reduce the reliance on labeled training data by minimizing the number of labeled examples required to learn classifiers. They typically do this by allowing the classifier to choose which example needs to be labeled next from a large pool of unlabeled examples, reducing supervision without sacrificing much accuracy in the model. The assumption is that while unlabeled examples can be collected with little or no effort, providing annotations on the examples entails non-trivial effort. Such methods are therefore appealing for object recognition because of the abundance of unlabeled images (available, for example, on the Web) and the substantial effort required to provide detailed annotations.

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

Department of Computer Science
University of Texas at Austin
E-mail: {svnaras,grauman}@cs.utexas.edu



(a) Most real-world images contain multiple objects and can therefore be associated with multiple labels.



(c) The actual manual effort required to label varies according to annotation type and image example.

(b) Useful image annotations can occur at multiple levels of granularity. For example, a learner may only know whether the image contains a particular object or not (top row, dotted boxes denote object is present), or it may also have segmented foregrounds (middle row), or it may have detailed outlines of object parts (bottom row).

Fig. 1 Three important problems that need to be addressed while choosing informative image data to label for recognition, none of which are considered by traditional active learning approaches.

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.

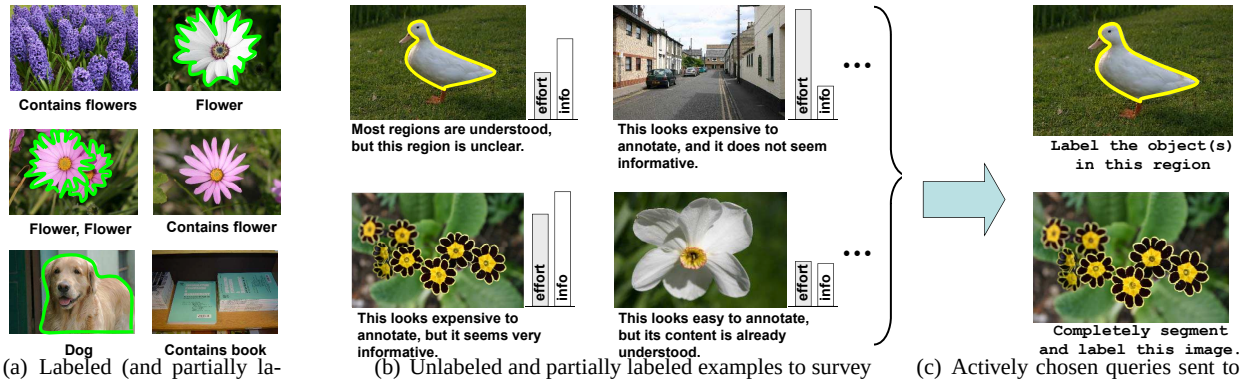
First, while many of today's manually collected datasets assume that the class to be learned occurs prominently in the foreground and therefore can be associated with a single label, most naturally occurring images consist of multiple objects. Therefore, an image can be associated with *multiple labels* simultaneously as shown in Figure 1(a). [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 (Figure 1(b)). Richer annotations such as segmentations provide more information from which to infer class membership, but require more effort on the part of the person providing supervision. While recent work has begun to explore ways to reduce the level of supervision (Weber et al (2000); Sivic et al (2005); Quelhas et al (2005); Bart and Ullman (2005); Fergus et al (2005); Li et al (2007); von Ahn and Dabbish (2004); Russell et al (2005); Verbeek and Triggs (2007)), such techniques fail to address a key issue: to use a fixed amount of manual effort most effectively may call for a combination of annotation at different supervision levels. Therefore, instead of ignoring annotations such as segmentations which require more effort to obtain, we need a principled way of predicting the tradeoff between the effort and information gain associated with any candidate image annotation. This means an active learner must be able to choose from annotations at multiple levels of granularity and specify not only which example but also what *type* of annotation is currently most helpful.

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. For example, completely segmenting an image and labeling all objects requires more time and effort than providing an image-level tag specifying object presence. 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, as seen in Figure 1(c)).

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 2). To accommodate the multiple levels of granularity that may occur in provided image annotations, we pose the problem in the multiple-instance learning setting (MIL). We show how to extend the standard binary MIL setting to the multi-label case by devising a kernel-based classifier for *multiple-instance, multi-label learning* (MIML). We formulate an active learning function in the MIML domain 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 labeled images, our method surveys any

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



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

(b) Unlabeled and partially labeled examples to survey

(c) Actively chosen queries sent to annotators

Fig. 2 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.

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. Humans (especially vision researchers) can easily glance at an image and roughly gauge the difficulty. 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. 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.

2 Related Work

A number of research threads aim at reducing the expense of obtaining well-annotated image datasets, from methods allowing weak supervision (Weber et al (2000)), to those that mine unlabeled images (Sivic et al (2005); Lee and Grauman (2008)). Other techniques reduce training set sizes by transferring prior knowledge (Fei-Fei et al (2003)), or exploiting noisy images from the Web (Fergus et al (2005); Vijayanarasimhan and Grauman (2008a)). Aside from such learning-based strategies, another approach is to encourage users to annotate images for free/fun/money (von Ahn and Dabbish (2004); Russell et al (2005); Sorokin and Forsyth (2008.)).

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 (Kapoor et al (2007a)), relevance feedback in video (Yan et al (2003)), dataset creation (Collins et al (2008)), and when there are correlations between image-level labels (Qi et al (2008)).

The multiple-instance learning (MIL) scenario has been explored for various image segmentation and classification tasks (Maron and Ratan (1998); Vijayanarasimhan and Grauman (2008a); Zhou and Zhang (2006)). Multi-label variants of MIL in particular are proposed in Zha et al (2008), with impressive results. Active selection in the two-class MIL setting was recently explored in Settles et al (2008), where the classifier is initially trained on examples with bag-level labels and active selection is performed to obtain instance-level labels on some examples. However, previous active learning 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, *and* labels at multiple levels of granularity. The multi-label distinction is important in practice, since in a naturally occurring pool of unlabeled data the images will not be restricted to containing only one prominent object. Similarly, the multi-level idea is

important since it will allow us to balance a mixture of annotation types.

In addition, most active learning approaches assume that each training example requires the same amount of manual effort resources to label. In reality, the effort involved in providing supervision could vary significantly depending on a number of factors. A theme is emerging in the learning community to quantify manual effort for different learning tasks. In Kapoor et al (2007b) and Baldridge and Osborne (2008), the length of a voice mail or sentence is used to identify examples that could take more or less manual effort to annotate. Budgeted learning for active classifiers, which work on constrained budgets while querying attributes on a test example, is explored in the work of Greiner et al (2002) for medical diagnosis. In Haertel et al (2008) regressors are learned based on sentence length, number of characters, etc. to predict annotation time for document classification. While the length of a voice mail or the cost of a medical diagnosis directly provides a measure of the cost of an example, no such direct measure exists to quantify the effort involved in providing an image annotation. Thus far, no existing approaches in object recognition attempt to quantify or predict the amount of effort required to provide annotations on image examples.

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 (Yan et al (2003); Kapoor et al (2007a); Collins et al (2008); Qi et al (2008)), and generally are limited to binary and/or single-label problems. 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.

This paper expands on our previous conference publications (Vijayanarasimhan and Grauman (2008b, 2009)); in this manuscript we provide a single framework to deal with both binary and multi-class problems. We provide additional results to demonstrate the robustness of our approach and illustrations to better understand the main ideas.

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.

We consider image collections consisting of a variety of supervisory information: some images are labeled as containing the category of interest (or not), some have both a class label and object outlines, while others have no annotations at all. We derive an active learning criterion function that predicts how informative further annotation on any particular unlabeled image or region would be, while accounting for the variable expense associated with different annotation types. Specifically, we show how to continually assess the value of three different types of annotations: a label on an image region, an image-level tag, and a complete segmentation of the entire image (see Figure 8). We also refer to these types as "levels", since they correspond to different levels of detail in the annotation. As long as the information expected from further annotations outweighs the cost of obtaining them, our algorithm will request the next valuable label, re-train the classifier, and repeat.

In the following, we introduce the MIL and MIML frameworks 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 multiple-instance learning (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 the MIL setting, as first defined by Dietterich et al (1997), 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. MIL is well-suited for the image classification scenario where training images are labeled as to whether they contain the category of interest, but they also contain background clutter. Every image is represented by a bag of regions, each of which is characterized by its color, texture, shape, etc. (Maron and Ratan (1998); Yang and Lozano-Perez (2000)). For positive bags, at least one of the regions contains the object of interest. The goal is to predict when new image regions

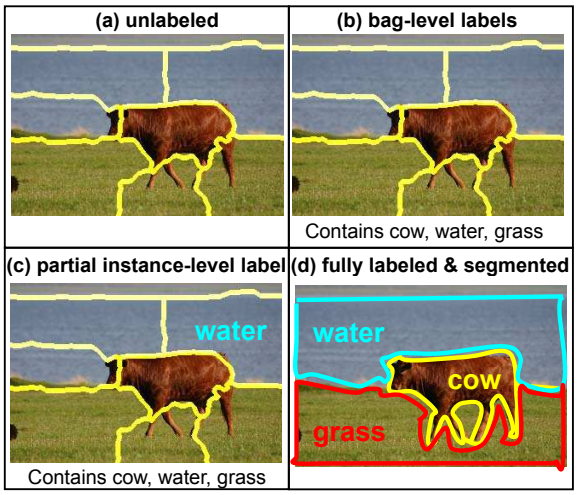


Fig. 3 In our MIML scenario, images are multi-label bags of regions (instances). Unlabeled images are oversegmented into regions (a). For an image with *bag-level* labels, we know which categories are present in it, but we do not know in which regions (b). For an image with some *instance-level* labels, we have labels on some of the segments (c). For a *fully annotated* image, we have true object boundaries and labels (d).

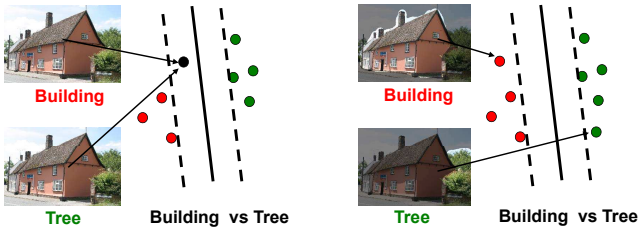


Fig. 4 The intuition behind our multi-label kernel function. **Left:** In MIML, if an image's representation is independent of its label, two different labels could map to the same point in feature space. **Right:** Our Multi-label Set Kernel weighs instances based on the predicted class membership, thereby associating specific regions within the image to the provided labels. In the top image the region containing a building (lighter shading) contributes more to the overall image representation given the label "building", while in the bottom image the region containing a tree contributes more for the label "tree".

contain the object—that is, to learn to label regions as foreground or background. Since a positive instance is a positive bag containing a single instance, MIL can accommodate both region labels (instance-level) and image tags (bag-level).

In the more general multiple-instance multi-label (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 $\{(X_1, L_1), (X_2, L_2), \ldots, (X_N, L_N)\}$ denote a set of training bags and their associated labels. Each bag consists of a set of instances $X_i = \{x_1^i, x_2^i, \ldots, x_{n_i}^i\}$, and a set of labels $L_i = \{l_1^i, l_2^i, \ldots, l_{m_i}^i\}$, 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_j^i is associated with a description $\phi(x_j^i)$ in some kernel embedding space and some class label $l_k^i \in \mathbb{L} = \{1, \ldots, C\}$, 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 3). 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. This follows the common use of MIL for images (Maron and Ratan (1998); Zha et al (2008); Vijayanarasimhan and Grauman (2008b)), 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: $\phi(X_i) = \frac{1}{|X_i|} \sum_{j=1}^{n_i} \phi(x_j^i)$, and then train an SVM with instances and bags; this is the Normalized Set Kernel (NSK) approach of Gartner et al (2002). The NSK is a kernel for sets, and is derived from the definition of convolution kernels using the set-membership function. In order to correct for the cardinality of the sets, a normalization factor based on the 1 or 2-norm is introduced. For the MIL setting, every instance in a bag can be seen as a member of the bag, and the NSK corresponds to an averaging process carried out in feature space. Bunescu and Mooney (2007) show that the NSK approach can be construed as a balancing constraint on the positive bags. Intuitively, this means that *on average* we expect the label on a positive bag to be greater than zero.

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 (see Figure 4 (left)). 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.

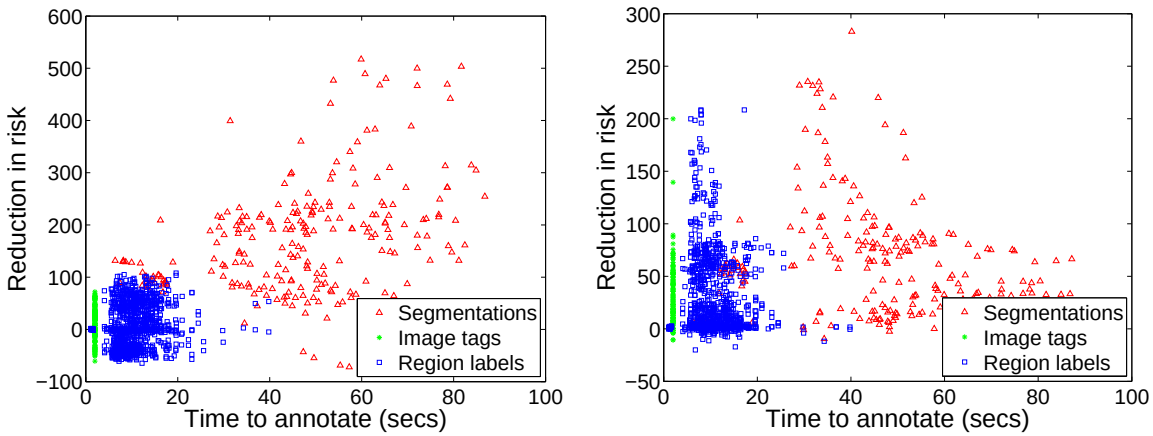


Fig. 5 This figure shows the reduction in risk for each example in the unlabeled pool plotted against the time required to provide an annotation after training with 5 image tags (**left**) and 100 image tags (**right**). There is not an absolute correlation between the cost of an annotation and how informative it is, motivating the use of cost-sensitive active learning.

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 (Figure 4 (right)). To this end, we design a class-specific feature representation of bags. Let $X = \{x_1, \ldots, x_n\}$ be a bag containing labels $L = l_1, \ldots, l_m$ (where here we drop the example index i for brevity). We define the class-specific feature vector of X for class l_k as

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

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 ⟨feature,label⟩ 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 (Gartner et al, 2002), which is restricted to single-label binary classification. It is also related to the kernel in (Kwok and Cheung, 2007), where weights are set using a Diverse Density function. In contrast, we estimate the class conditional probabilities using the classifier constructed with the currently available training data.

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 the method of Platt (1999), 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 Wu et al (2004), where the pairwise class probabilities are used to solve a linear system of equations to obtain the multi-class probabilities.

While in this paper we have combined one-vs-one binary problems to obtain a multi-class classifier, our method is not restricted to this setting. Since our approach defines a kernel for the multi-label problem, it can be used with other kernel-based multi-class approaches, including one-vs-all SVMs.

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; the most cost-effective queries are those where informativeness outweighs effort. 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. Figure 8 illustrates each of these three types.

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 illustrate this idea more concretely, we ran an experiment where we measured both the reduction in misclassification risk produced by adding an annotation with its correct label from an unlabeled pool of images and the time to obtain the annotation. The misclassification risk is defined in the standard way, as the probability of classifying each example with an incorrect label, summed over all examples. Figure 5 shows this result for all examples in the unlabeled pool with three annotation types (segmentations, image tags and region labels) for two different sizes of the initial training set (5 and 100 image tags respectively).

The figures suggest that neither more expensive nor less expensive examples are regularly more useful than the other. Similarly, the annotation that provides the best reduction in risk might not be the most effective in terms of the cost of obtaining it. For example, in Figure 5 (right) there are examples from all three annotation types with reductions in risk above 200 units. While a standard "flat cost" active learner would choose the more expensive segmentation (because of the marginally higher reduction in risk) a cost-sensitive learner might choose the less expensive one.

The figures also illustrate that while segmentations are indeed more expensive to obtain, the larger reductions in risk can effectively mitigate the cost for several examples. In addition, the relative risk reduction versus the annotation time required is a function that continually changes as more annotated data is acquired, as evident when we compare the total shape of the scatter plots on the left (where only 5 examples have been seen per class) and on the right (where 100 examples have been seen per class). Hence, to best reduce human involvement, the active learner needs a quantitative measure of the effort required to obtain any given annotation.

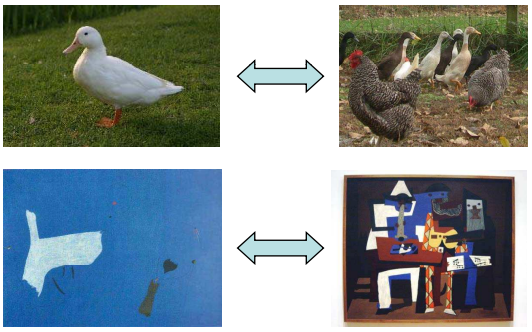


Fig. 6 Which image would you rather annotate? Humans can easily glance at an image and roughly gauge the difficulty. This appears to be true even without prior knowledge about the specific objects present in the image (second row).

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. As Figure 6 suggests, humans are able to predict the difficulty of annotating an image even without prior knowledge about the objects occurring in the image (second row) or other high-level cues. Therefore, it seems plausible that the difficulty level of an image could be predicted based on the image's low-level 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 to select those most useful for the task. Multiple kernel learning approaches automatically select the weights on the various features (kernels) by posing the problem as an optimization of the coefficients of such a combination. Lanckriet et al (2004) show that this reduces to a convex optimization problem known as a quadratically-constrained quadratic program (QCQP). Bach et al (2004) propose a novel dual formulation of the corresponding QCQP as a second-order cone programming problem to yield a formulation for which the sequential minimal optimization (SMO) algorithm

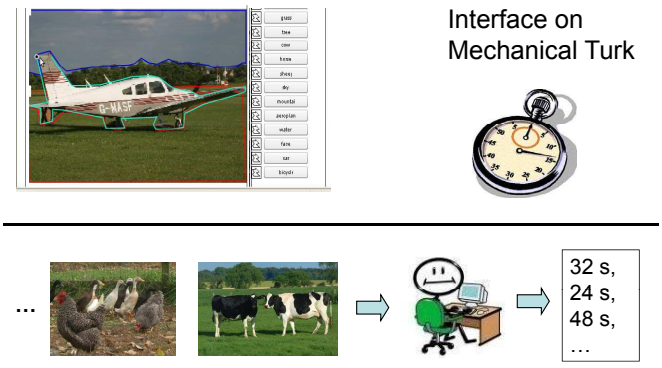


Fig. 7 Our interface on Mechanical Turk to collect annotation times for segmenting images from anonymous users. The system times the responses as users use a polygon-drawing tool to superimpose object boundaries, and name and outline every major object.

can be applied. We use this SMO algorithm to select cost-predictive features, since it allows efficient solutions for large-scale problems.

We begin with some generic features that may be decent indicators of image complexity: a histogram of oriented gradients, a gray-scale histogram, and two new features based on the edge density and color uniformity. The features are designed to exploit the fact that more objects could lead to more annotation time.

The edge density feature divides the image into a hierarchical grid of cells and concatenates the edge density within each cell into a feature vector. We reason that edge density could be a good indicator of the number of objects, since with a larger number of objects in an image there are bound to be more edges separating them. The hierarchy, by capturing edge densities at multiple scales, helps in dealing with objects of different scales.

The color uniformity feature computes the standard deviation of the r, g, b values of every pixel in the image based on a small neighborhood surrounding it, and obtains a histogram of the standard deviations. With more objects we expect larger standard deviations in a neighborhood compared to a small number of smoothly varying regions such as sky, grass, etc.

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 (see Figure 7). 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.

We construct a χ^2 RBF kernel over the training examples per image feature. Based on the timing obtained from the anonymous users we divide the set of training images into a discrete range of "easy" and "hard" images using the mean time over all the images. We then use the MKL approach of Bach et al (2004) to learn the weights on the image features for the binary classification problem of classifying images into "easy" and "hard" categories. Using the obtained combined kernel, we also learn a cost predictor function using Support Vector Regression (SVR).

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 (Kapoor et al (2007b)), 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. In contrast to previous VOI methods, our measure enables the multi-label setting and considers multiple types of annotations to select from.

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 (refer back to Figure 3). If the label on an image is considered to be a binary vector of length C, then the images in $\mathcal{X}_L$ are examples where the binary label vector is completely known. Images in $\mathcal{X}_U$ are examples where none of the labels are known, and images in $\mathcal{X}_P$ are examples where some of the elements in the vector and labels on some of its instances are known. An example is moved from $\mathcal{X}_U$ to $\mathcal{X}_P$ when any one of its unknown labels is requested. An example is moved from $\mathcal{X}_P$ to $\mathcal{X}_L$ only when the labels on all its instances have been obtained.

(a) Name an object in the image (unlabeled bag).

(b) Label the specified region (unlabeled instance).

(c) Segment the image and name all objects (label all instances).

Fig. 8 The three candidate annotation types (or "levels") that our approach chooses from when formulating a request.

Let r_l denote the risk associated with misclassifying an example belonging to class l. The risk associated with $\mathcal{X}_L$ is:

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

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 compute the probability it receives label l as follows:

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

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 MIML classifier implicitly assumes that every image/instance can be classified into one of C labels. However, in the more general case, the dataset can also contain images that do not necessarily belong to the C classes. Such images are given a "negative" label which specifies that none of the instances/regions in the image belong to any of the classes in $\{1, \ldots, C\}$, similar to the "negative" label in a standard MIL formulation. In this case, we weight $p(l|X_i)$ with the probability of X_i belonging to any one of the C classes as against the "negative" class, which is obtained by training a standard MIL classifier. Note that when $C = 1$, a single foreground class, the above reduces to the standard MIL solution since $p(l|X_i)$ is trivially 1. Similarly, in the absence of a "negative" class the above reduces to the MIML solution.

The corresponding risk for the unlabeled data is:

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

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:

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

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

The value r_l is the risk associated with misclassifying an example belonging to class l, specified in the same units as the cost function in Section 3.2.1. Intuitively, it should reflect the real cost of a classification mistake, as our algorithm directly trades off the cost of the manual labeling against the damage done by misclassification. While this can be set based on realistic system requirements, we interpret it as the cost of manually fixing a classification error (e.g., an average segmentation requires 50 secs). If one preferred to avoid errors on a particular class, that could be encoded with variable r_l values per class label l. Note that r_l is not a parameter that needs to be optimized for performance; rather, it gives flexibility for situations that have real costs associated with the task.

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:

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

where $\mathcal{X}_B = \mathcal{X}_L \cup \mathcal{X}_P$, and $\mathcal{C}(\cdot)$ is defined in Section 3.2.1.

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:

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

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 $\mathbf{z}$. Note that $\mathbf{z}$ could be any one among the three annotation types described in Figure 8. If all the labels on the example have been obtained through $\mathbf{z}$ then the example is moved to the labeled pool, i.e., $\hat{\mathcal{X}}_L = \mathcal{X}_L \cup \mathbf{z}$. On the other hand, if the example contains instances (regions) with no label information even after obtaining annotation $\mathbf{z}$ then the example is moved to the set of partially labeled data, i.e., $\hat{\mathcal{X}}_P = \mathcal{X}_P \cup \mathbf{z}$. Similarly, the example associated with $\mathbf{z}$ is removed from $\mathcal{X}_U$ or $\mathcal{X}_P$ 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: $\mathcal{R}(\hat{\mathcal{X}}_L) + \mathcal{R}(\hat{\mathcal{X}}_U) + \mathcal{R}(\hat{\mathcal{X}}_P) \approx E[\mathcal{R}(\hat{\mathcal{X}}_L) + \mathcal{R}(\hat{\mathcal{X}}_U) + \mathcal{R}(\hat{\mathcal{X}}_P)]$, henceforth denoted by $\mathbb{E}$.

If the annotation $\mathbf{z}$ will label an unlabeled instance (Figure 8(b)), computing the expectation is straightforward, since that instance can simply be removed from $\mathcal{X}_U$ and added to $\mathcal{X}_L$ to evaluate the risk were it assigned each of the L possible labels in turn:

$$\mathbb{E} = \sum_{l \in \mathbb{L}} \Big(\mathcal{R}(\mathcal{X}_L \cup \mathbf{z}^{(l)}) + \mathcal{R}(\{\mathcal{X}_U, \mathcal{X}_P\} \setminus \mathbf{z}) \Big) \Pr(l|\mathbf{z}), \tag{7}$$

where $\mathbb{L} = \{1, \ldots, C\}$ is the set of all possible label assignments for $\mathbf{z}$. The value $\Pr(l|\mathbf{z})$ is obtained by evaluating the current classifier on $\mathbf{z}$ and mapping the output to the associated posterior, and risk is computed based on the (temporarily) modified classifier with $\mathbf{z}^{(l)}$ inserted into the labeled set. Similarly, if the candidate annotation $\mathbf{z}$ will add an image-level label to an unlabeled or partially labeled bag (Figure 8(a)), then $\Pr(l|\mathbf{z})$ is calculated using Eqn. 3.

However, if the annotation $\mathbf{z}$ entails fully segmenting and labeling an image with M automatically segmented regions (Figure 8(c)), we need to calculate the utility of obtaining the joint set of labels for all of a bag's instances. Since there are C^M possible labelings: $\mathbb{L} = \{1, \ldots, C\}^M$, a direct computation of the expectation is impractical. Instead we use Gibbs sampling to draw samples of the label assignment from the joint distribution over the M instances' descriptors. Let $\mathbf{z} = \{z_1, \ldots, z_M\}$ be the bag's instances, and let $\mathbf{z}^{(\mathbf{a})} = \left\{ (z_1^{(a_1)}), \ldots, (z_M^{(a_M)}) \right\}$ denote the label assignment we wish to sample, with $a_j \in \{1, \ldots, C\}$. To sample from the conditional distribution of one instance's label given the rest—the basic procedure required by Gibbs sampling—we re-train the classifier with the given labels added, and then draw the remaining label according to $a_j \sim \Pr(l|z_j)$, for $l \in \{1, \ldots, C\}$, where z_j denotes the one instance currently under consideration. For bag $\mathbf{z}$, the expected total risk is then the average risk computed over all samples:

$$\mathbb{E} = \frac{1}{S} \sum_{k=1}^{S} (\mathcal{R}(\{\mathcal{X}_L \setminus \mathbf{z}\} \cup \{z_1^{(a_1)_k}, \ldots, z_M^{(a_M)_k}\}) + \mathcal{R}(\mathcal{X}_U \setminus \{z_1, z_2, \ldots, z_M\}) + \mathcal{R}(\mathcal{X}_P)), \tag{8}$$

where k indexes the S samples. We compute the risk on $\mathcal{X}_L$ for each fixed sample by removing the bag $\mathbf{z}$ from the unlabeled or partially labeled pool, and inserting its instances with the label given by the sample's label assignment. Note that while computing the VOI of a candidate annotation we have no supervision information on that example, including the object outlines. Hence, the computation of VOI is performed using segments/regions generated using an automatic segmentation algorithm. Once we obtain a complete segmentation of an image from the annotator, we use the actual region outlines and labels to retrain the classifier.

Computing the VOI values for all unlabeled data, especially for the positive bags, requires repeatedly solving the classifier objective function with slightly different inputs; to make this manageable we employ incremental SVM updates (Cauwenberghs and Poggio, 2000).

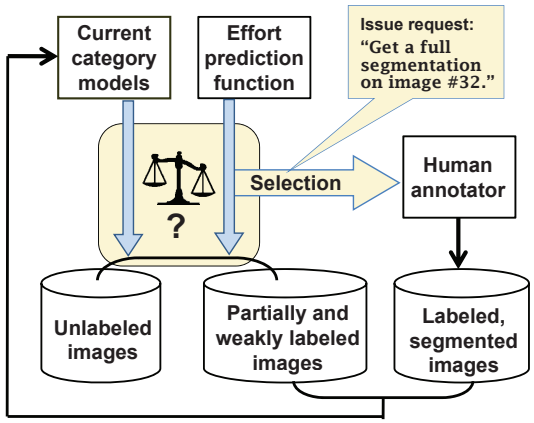


Fig. 9 The summary of our approach. After learning from a small initial set of labeled images, our method surveys any available unlabeled and partially labeled data. The VOI of every candidate annotation among three different types of annotations is computed using the expected change in risk and the predicted effort of obtaining the annotation given by our cost predictor. The annotation and example with the largest VOI is then selected and a human provides the annotation, after which the example is moved from the unlabeled/partially labeled pool to the partially/fully labeled pool as appropriate. The process repeats until there are no more examples with positive VOI, or once the allowed annotation cost limit has been reached.

Fig. 10 Example images from the SIVAL dataset. Each row illustrates one of the 25 objects.

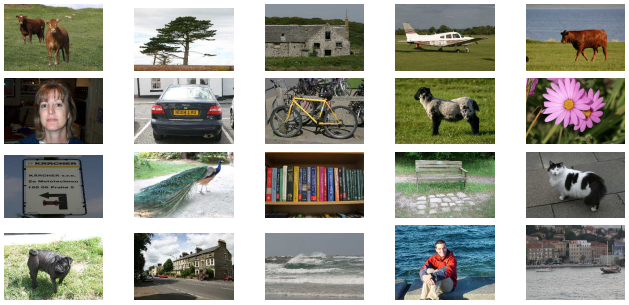


Fig. 11 Example images from the MSRC dataset. The MSRC dataset contains 21 categories, and most images have multiple categories in them (eg: "building", "road", "sky", "tree").

Approach	Ave. AUROC (img)	Ave. AUROC (region)
Ours	0.896 ± 0.00	0.91 ± 0.01
MLMIL (Zha et al, 2008)	0.902	0.863

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

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. Figure 9 provides a high-level summary of the approach. The final classifier can predict image- and region-level labels, in binary or multi-class settings.

4 Results

To validate our method we use two publicly available datasets, the SIVAL[2] dataset and the MSRC [3] dataset, since they have been used to evaluate previous MIL and MIML based approaches which allows us to compare with state-of-the-art methods in the two settings. Additionally, the MSRC is a common benchmark for multi-class object segmentation.

The SIVAL dataset contains images from 25 objects in cluttered backgrounds, while the MSRC v2 contains 591 images from 21 classes and a variable number of objects per image, with 240 images and 14 classes in the (subset) v1. See Figures 10 and 11 for examples. In all MSRC experiments we use an RBF kernel with $\gamma = 10$, and set the SVM parameters (including the sigmoid parameters for the SVM probabilistic outputs given by the method of (Platt, 1999)) based on cross-validation. We ignore all "void" regions in the MSRC images.

In the following subsections, we evaluate five aspects of our approach: (1) its accuracy when learning from multi-label examples, (2) its ability to accurately predict annotation costs, (3) its effectiveness as an active learner when selecting from three different types of annotations on both binary and multi-label problems, (4) the effect of introducing the cost predictor in the active selection function, and (5) the robustness of our approach with respect to the initial training set.

4.1 Multi-label visual category learning with the MSK

In our first experiment, we evaluate our proposed multi-label set kernel classifier's effectiveness in learning using only image-level labels on images containing multiple objects. We divide the MSRC v2 into five folds containing about an equal number of images, as is done by Zha et al (2008). We choose one part as the test set, one to set parameters, and train on the rest. We segment the images with Normalized Cuts into a small number of segments (10 in our experiments). For each segment we then obtain texton and color histograms, as in (Shotton et al, 2006). We learn a dictionary of textons by convolving the images with a 38-dimensional filter bank and running K-means clustering to obtain 420 textons. For color histograms we obtain a 120-dimensional vector by concatenating a 40-dimensional histogram of each channel of the LUV representation of the image.

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. We first compare against the approach of Zha et al (2008), who provide state-of-the-art results on the MSRC dataset while learning from image-level labels.

Table 1 shows the average AUROC when predicting labels on new *images* (second column) or new *regions* (third

[2] http://www.cs.wustl.edu/accio/

[3] http://research.microsoft.com/en−us/projects/objectclassrecognition/

column). For image-level prediction our results are comparable to the state-of-the-art in MIML (Zha et al (2008)), 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.

Next we compare against the approaches of Shotton et al (2006) and Winn et al (2005), 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 five trials of approximately equal train-test splits. In comparison, the accuracy obtained for the same test scenario is 70.5% in Shotton et al (2006), and 67.6% in Winn et al (2005). Using both region- and bag-level labels we obtain an accuracy of 66.3%. 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 in effectively utilizing weaker supervision.

Finally, using the NSK (Gartner et al (2002)), which essentially removes our kernel weight mapping, the accuracy for this test would only be $55.95\% \pm 1.43$. This result indicates that the proposed method to map different regions to the image-level labels is more effective.

4.2 Actively learning visual objects and their foreground regions

In this section we demonstrate our approach to actively learn visual categories for both the binary setting, where an image contains a single object of interest in a cluttered background as well as the multi-label setting, where an image contains multiple familiar objects that must be segmented and classified. We test both datasets described above.

We provide results by simulating the active learning process: when the system requests an annotation on an image example, we satisfy the request using the ground truth labels. For instance, when the request is to outline all the objects in the image, we use the ground truth segmentation provided with the dataset (SIVAL/MSRC) to obtain all the objects and their labels. However, recall that when calculating the VOI of a region/image, the system uses an automatic low-level segmentation of the image.

4.2.1 Active selection from MIL data

For the binary MIL setting, we provide comparisons with single-level active learning (with both the method of Settles et al (2008), and where the same VOI function is used but is restricted to actively label only instances), as well as passive learning. For the passive baseline, we consider random selections from amongst both single-level and multi-level annotations, in order to verify that our approach does not simply benefit from having access to more informative possible labels.

The SIVAL dataset contains 1500 images, each labeled with one of 25 class labels. The cluttered images contain objects in a variety of positions, orientations, locations, and lighting conditions. The images have been oversegmented into about 30 regions (instances) each, each of which is represented by a 30-d feature capturing the average color and texture values of the segment and each of its cardinal neighbors. These features are provided with the SIVAL dataset [4]. Thus each image is a bag containing both positive and negative instances (segments). Labels on the training data specify whether the object of interest is present or not, but the segments themselves are unlabeled (though the dataset does provide ground truth segment labels for evaluation purposes).

For Gibbs sampling, we generate $S = 25$ samples with an initial burn-in period of 50 samples. This number was set arbitrarily; later experiments increasing the sample size to 50 did not improve results significantly, though in general larger samples should yield more accurate VOI estimates. The risk parameter (r_l) and the cost of labeling a single instance are all set to 1, meaning we have no preference for false positives or false negatives, and that we view a misclassification to be as harmful as requiring a user to label one instance.

As the SIVAL dataset contains exactly one object per image (see Figure 10), we do not expect the segmentation costs to vary on a per example basis. Therefore, for this dataset we attribute a single cost to all annotations of a particular type. To determine how much more labeling a positive bag costs relative to labeling an instance, we performed a user study. Users were shown oversegmented images and had to click on all the segments belonging to the object of interest. The baseline task was to provide a present/absent flag on the images. For segmentation, obtaining labels on all positive segments took users on average four times as much time as setting a flag. Thus we set the cost of labeling a positive bag to 4 for the SIVAL data. The value agrees with the average sparsity of the dataset: the SIVAL set contains about 10% positive segments per image. The users who took part in the experiment were untrained but still produced consistent results.

The initial training set is comprised of 10 positive and 10 negative images per class, selected at random. Our active learning method must choose its queries from among 10 positive bags (complete segmentations), 300 unlabeled

[4] http://www.cs.wustl.edu/∼sg/accio/SIVAL.html

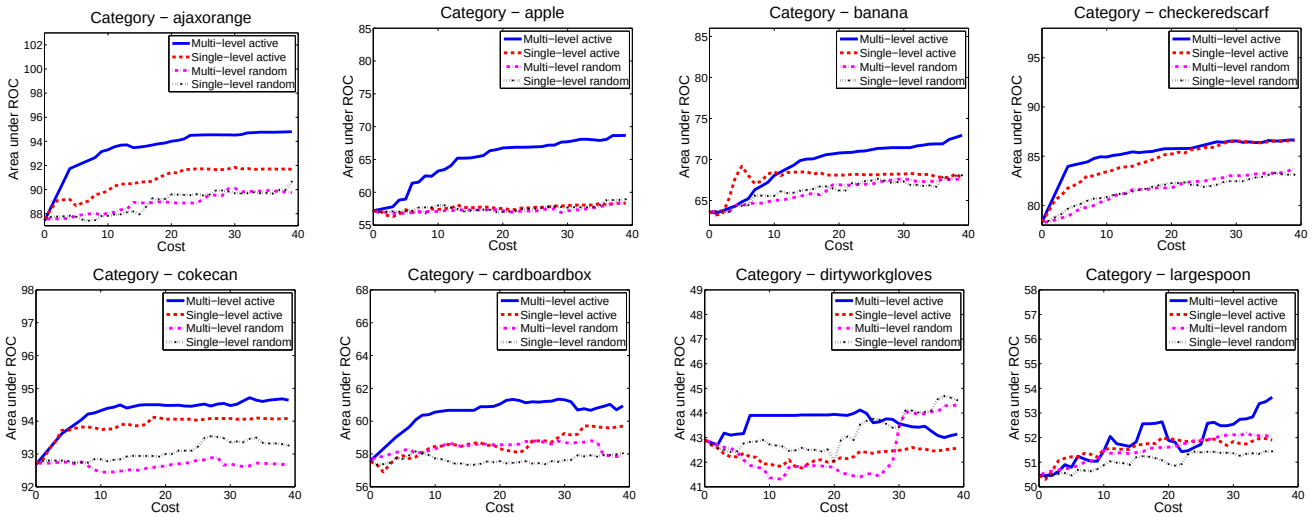
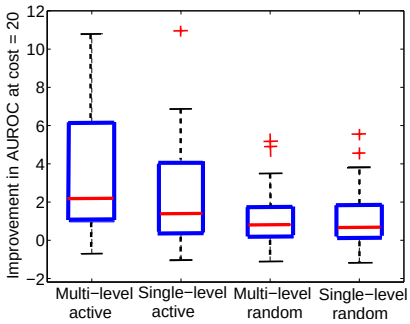


Fig. 12 Results on the SIVAL dataset. Sample learning curves per class, each averaged over five trials. Our method corresponds to the "Multi-level active" curves. First six are best examples, last two are worst. For the same amount of annotation cost, our multi-level approach learns more quickly than both traditional single-level active selection as well as both forms of random selection.



Cost	Our Approach			Settles et al (2008)		
	Random	Multi-level Active	Gain over Random (%)	Random	MIU Active	Gain over Random (%)
10	+0.0051	+0.0241	372	+0.023	+0.050	117
20	+0.0130	+0.0360	176	+0.033	+0.070	112
50	+0.0274	+0.0495	81	+0.057	+0.087	52

Fig. 13 Left: Summary of the average improvement over all 25 SIVAL categories after half of the annotation cost is used. **Right:** Comparison with Settles et al (2008) on the SIVAL data, as measured by the average improvement in the AUROC over the initial model for increasing labeling cost values.

instances (individual segments), and about 150 unlabeled bags (present/absent flag on the image). We use a quadratic kernel, $K(x, y) = (1 + \alpha\phi(x)^T\phi(y))^2$, with a coefficient of $\alpha = 10^{-6}$, and average results over five random training partitions.

Figure 12 shows representative (best and worst) learning curves for our method and the three baselines, all of which use the same MIL classifier (NSK-SVM). Note that the curves are plotted against the cumulative *cost* of obtaining labels—as opposed to the number of queried instances—since our algorithm may choose a sequence of queries with non-uniform cost. All methods are given a fixed amount of manual effort (40 cost units) and are allowed to make a sequence of choices until that cost is used up. Recall that a cost of 40 could correspond, for example, to obtaining labels on $\frac{40}{1} = 40$ instances or $\frac{40}{4} = 10$ positive bags, or some mixture thereof. Figure 13 (left) summarizes the learning curves for all categories, in terms of the average improvement at a fixed point midway through the active learning phase.

All four methods steadily improve upon the initial classifier, but at different rates with respect to the cost. (All methods fail to do better than chance on the 'dirty glove' class, which we attribute to the lack of distinctive texture or color on that object.) In general, a steeper learning curve indicates that a method is learning most effectively from the supplied labels. Our multi-level approach shows the most significant gains at a lower cost, meaning that it is best suited for building accurate classifiers with minimal manual effort on this dataset. As we would expect, single-level active selections are better than random, but still fall short of our multi-level approach. This is because single-level active selection can only make a sequence of greedy choices while our approach can jointly select bags of instances to query. Interestingly, multi- and single-level random selections perform quite similarly on this dataset (see boxplots in Figure 13 (left)), which indicates that having more unambiguous labels alone does not directly lead to better classifiers unless the right instances are queried.

At a cost of 24 units the mean AUROC over all 25 classes for active selection turned out to be 0.723, which is 92% of the accuracy achievable if using *all* the labels and examples in the unlabeled pool. To reach the same accuracy random selection requires 44 units of cost. This means that to reach 92% of the upper-bound accuracy, active selection requires 45.5% less annotation cost than the passive learner.

The table in Figure 13 compares our results to those reported in (Settles et al, 2008), in which the authors train an initial classifier with *multiple-instance logistic regression*, and then use the MI Uncertainty (MIU) to actively choose instances to label. To our knowledge this is the only other existing approach to perform active selections with MIL data, making it a useful method to compare to. Following Settles et al (2008), we report the average gains in the AUROC over all categories at fixed points on the learning curve, averaging results over 20 trials and with the same initial training set of 20 positive and negative images. Since the accuracy of the base classifiers used by the two methods varies, it is difficult to directly compare the gains in the AUROC. The NSK-SVM we use consistently outperforms the logistic regression approach using only the initial training set; even before active learning our average accuracy is 68.84, compared to 52.21 in (Settles et al, 2008). Therefore, to aid in comparison, we also report the percentage gain relative to random selection, for both classifiers. The results show that our approach yields much stronger relative improvements, again illustrating the value of allowing active choices at multiple levels (the method of Settles et al (2008) only allows active queries for instance-level labels). For both methods, the percent gains decrease with increasing cost; this makes sense, since eventually (for enough manual effort) a passive learner can begin to catch up to an active learner.

While these results illustrate the MIL scenario where images are bags of regions, our approach is applicable for any scenarios where there are two label granularities. In a previous paper (Vijayanarasimhan and Grauman (2008a)), we introduced another image-classification scenario for which MIL is well-suited, where the keyword associated with a category is used to download groups of images from multiple search engines in multiple languages. Each downloaded group is a bag, and the images within it are instances. For each positive bag, at least one image actually contains the object of interest, while many others may be irrelevant. The goal is to predict the presence or absence of the category in new images. See Vijayanarasimhan and Grauman (2008b) for active selection results for this alternate MIL scenario.

4.2.2 Active selection from MIML data

In the previous section we considered active selection in the binary setting when the image contains a single object among background clutter. Next we use the MSRC dataset to demonstrate the impact of using our multi-label active selection function in the more general multi-label setting, where an image contains multiple objects of interest plus clutter, and selections can be made from different types of annotations.

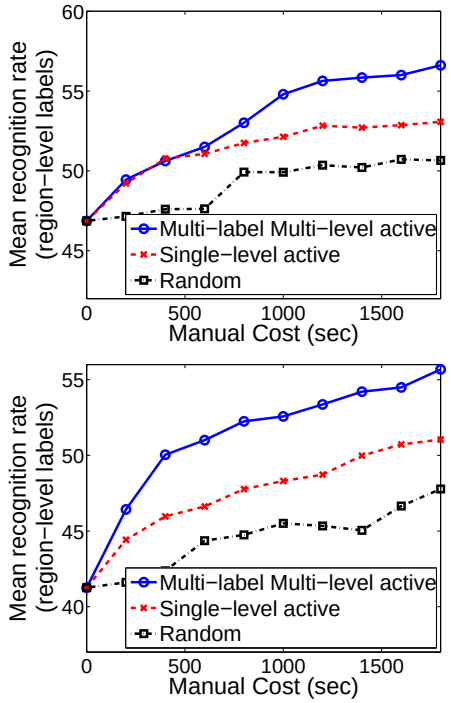
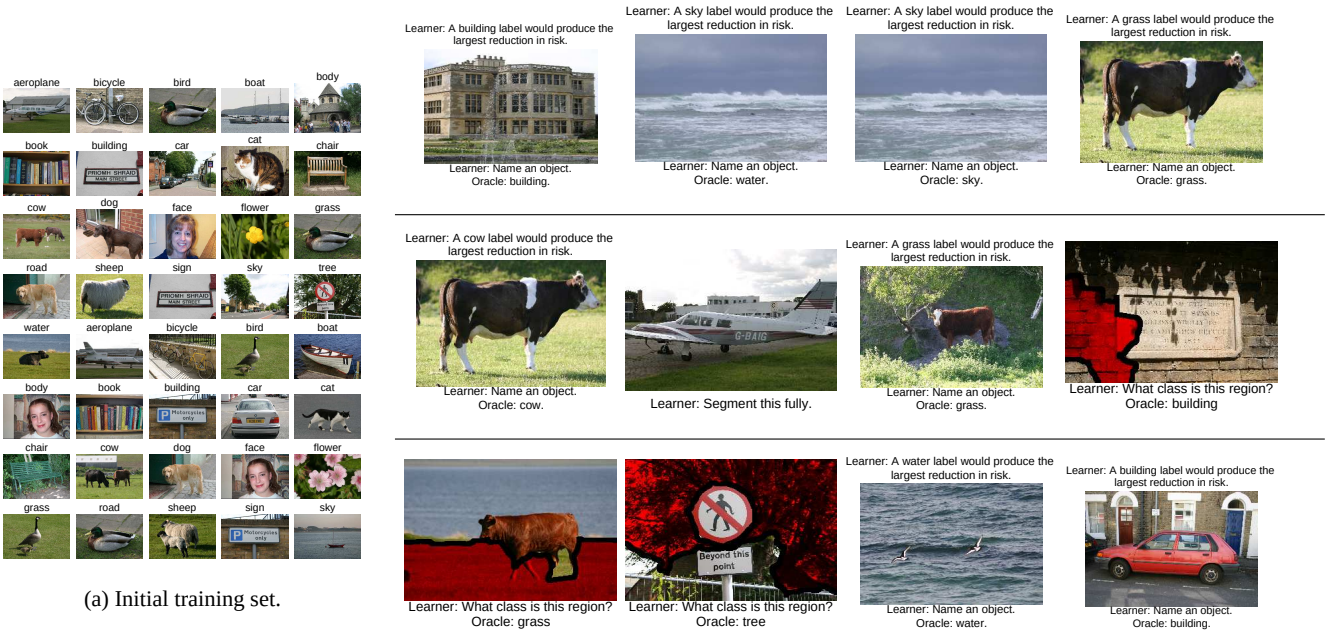


Fig. 14 Learning curves when actively or randomly selecting multi-level and single-level annotations. **Top:** Region-level accuracy for the 21-class MSRC v2 dataset plotted against ground truth cost. **Bottom:** Region-level accuracy when 80 random images were added to the unlabeled pool. Our multi-level active selection approach yields the steepest learning curves while random selection lags behind, wasting annotation effort on less informative examples. When 80 random images are added to the unlabeled pool, random selection lags even further, since there are more uninformative images that it can choose.

We divide the examples into five folds containing an equal number in each and use the first part for training and the rest for testing. We construct the initial training set such that each class appears in at least five 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. The parameter r_l should reflect the real cost of a classification mistake. Our choice of the value of r_l is based on the fact that an error made by the automatic labeling would take around 50 *s* to manually fix for the average image. 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"



(a) Initial training set.

(b) Annotations selected by the active learner in order (row major).

Fig. 15 Annotation queries selected by our method (right) on an example run starting from a small training set containing two examples per class (left). **Right:** Each image (from left to right) represents the example with the largest VOI as selected by our active learner on a sequence of iterations. The active learning query (one among a region label, an image tag, or a complete segmentation) is displayed at the bottom of the image along with the oracle's answer. For a query on a region, the corresponding region is highlighted in the image; for an image tag, the text on the top of the image represents what label is expected to produce the best reduction in risk.

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 until the cost is used up.

Figure 14 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 five random trials. The proposed multi-level active selection yields the steepest learning curves. Random selection lags behind, wasting annotation effort on less informative examples. As before, 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 1800 secs 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 14 (bottom) shows results for the same setting when 80 random images are added to the unlabeled pool with the "negative" class label, indicating that the more uninformative images that are present, the more random selection will lag behind.

When active and random selection are run to completion on all labels, both methods reach an accuracy of 59.5% [5]; random selection requires 5776 units of manual effort to reach the upper-bound while active selection requires only 3075 units. Thus with active selection we reach the upper bound using 46.7% less cost than the passive learner requires.

4.2.3 Active selection examples

In this section we look at the types of annotation queries that our approach requests based on some qualitative and quantitative results. Figure 15 shows annotation queries selected by our approach during the first 12 iterations of an example run starting from a small training set consisting of two image tags per class. The initial training set is displayed in Figure 15(a), and Figure 15(b) shows the first 12 queries selected by our approach in row major order. The type of query and the result from the oracle are displayed at the bottom of the image. We also highlight the region being queried in the case of a region label; text on the top of the image

[5] Note that since we use a different train-test split for experiments in this section, this upper-bound is not comparable to the accuracy reported in Section 4.1

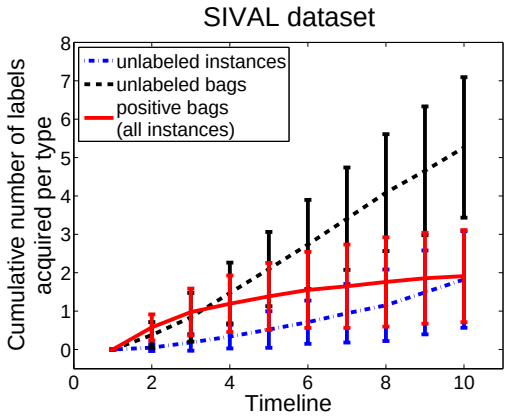


Fig. 16 The cumulative number of labels acquired for each type with increasing number of queries. Our method tends to request complete segmentations or image labels early on, followed by queries on unlabeled segments later on. This agrees with the intuition that fewer segmentations are worth their higher annotation costs as the classifier becomes stronger.

shows which image tag our approach thinks would produce the biggest reduction in the risk (the l with the largest value in the summation in Equation 7).

The annotations requested by our approach are dominated by image tags, which is reasonable considering they are the least expensive labels among the three types. At the same time, the images for which tags are requested appear to consist of a small number of clearly defined objects ('sky', 'water' in the second and third images, 'water', 'building' in the first image, etc.). On more complex images, such as the sixth image of the airplane, a complete segmentation is requested. Also a region label on the 'tree' region is requested on the tenth image, even though a tree image tag is already available on the same image in the training set. This illustrates that in some cases stronger annotations might be required, even when the classifier already contains weaker information about a class.

The examples selected by our approach are also diverse in their appearance and class labels. For example, in the images selected by our approach that contain the region 'sky', the appearance of the region is distinct from the examples of 'sky' already available in the training set. This is also the case for classes 'building' and 'water'.

Figure 16 shows the cumulative number of labels acquired for each type of annotation with increasing number of queries on the SIVAL dataset for the case of binary classification. Our previous observation on the larger proportion of image tags holds true in this dataset too. In addition, on this dataset our approach appears to select complete segmentations early on, followed by queries on unlabeled segments later on. Intuitively, as the classifier becomes stronger it may be that fewer segmentations can provide adequate risk reductions to mitigate their higher costs, and hence the less expensive image tags become favorable.

4.2.4 Effect of initial training set size

A well-known concern when performing active selection is that a faulty initial model might select uninformative examples to label and thus never converge to the most general hypothesis. Thus, we next consider the robustness of our approach by varying the number of training labels used to train the initial classifier. For the MSRC dataset we train the initial classifier with two, four, and eight image tags per class (42, 84, and 125 image tags overall) and then perform active selection with each model. In Figure 17, we compare our multi-level active selection approach against a multi-level random baseline and the best possible selection criterion. The best possible selection is obtained by computing the actual VOI of an example using its ground truth label. This is to compare how closely our expected VOI can approximate the actual VOI. We average results over five random trials.

On all three initializations, particularly for the smaller sets, our active selection approach has a larger slope than random selection. In addition, our active selection follows the trend of the best possible selection criterion. This illustrates the robustness of the approach to the initialization on this particular dataset. Also, since our multi-class classifier is an ensemble of a large number of binary classifiers, even with two image tags per class the final classifier could have enough examples to discriminate between the classes.

We show results for the same experiment for binary classification on the SIVAL dataset in Figure 18. The figure shows some representative (best and worst) learning curves comparing our selection function and a random baseline starting with two, six and twenty examples equally distributed across the positive and negative classes. The results are averaged over six random trials. Note that the three curves start at different points on the cost axes because they start with a different number of training examples. However, accuracies at a particular cost on the different curves are not necessarily comparable since the random initialization selects an equal number of positive and negative examples, while active and random selection approaches select from an unbalanced pool of positive and negative examples due to the one-vs-all binary setting.

The more variable results, as seen in the figure, could point to a harder dataset or the extremely low number of examples used in the binary setting as compared to the multi-class setting. The first row of learning curves show examples where a good initialization (larger number of examples) helps the active selection criterion. On these examples it appears that with smaller number of examples the active selection criterion could be misled into regions of the hypothesis space that do not necessarily correspond to the most general solution for the given training set.

The first two curves in the second row are examples where even with very few training examples the active selec-

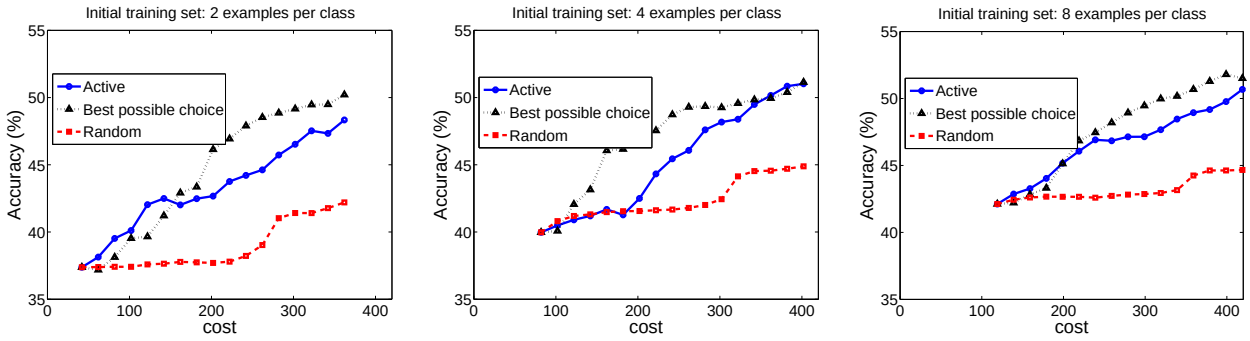


Fig. 17 Effect of the initial training set size on the active selection on the MSRC dataset. The classifier is initialized with two (left), four (middle), and, eight (right) image tags per class, and active selection is compared with a random baseline and the best possible selection criterion based on the actual VOI. On the MSRC dataset our active selection criterion is robust to the initialization and performs much better than random selection on all three initial training sets.

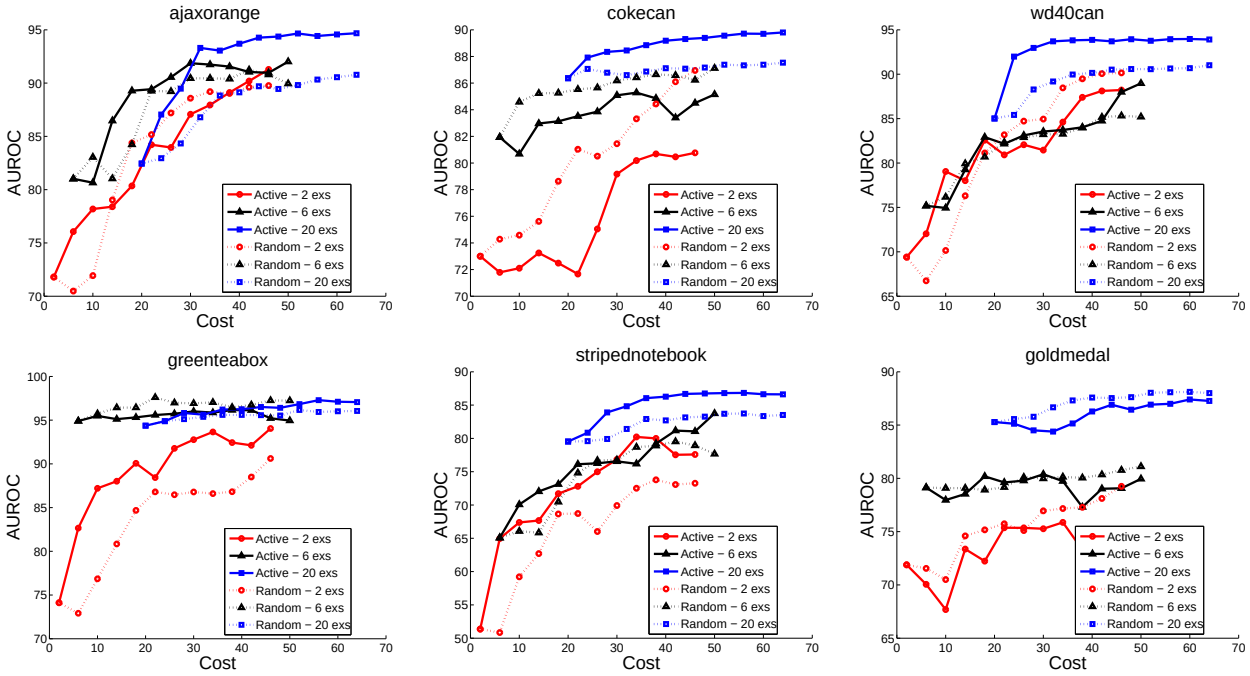


Fig. 18 Effect of the initial training set size on active selection on the SIVAL dataset. We initialize the classifier with two, six, and twenty image tags equally distributed across positive and negative classes. The figure shows some representative (best and worst) learning curves for our active selection approach and a random baseline. On this dataset a small training set composed of only two examples produces sub-optimal selections for some classes.

tion criterion is able to match results with a larger initial set. The final curve in the second row shows an example where active selection performs worse than random on all three initializations. These results suggest that active learning could be affected by the initialization on certain problems. However, note that we deliberately chose an extremely small initial training set (two, six examples) to illustrate this point. Arguably, for most real applications one can reasonably expect to initialize the model with at least 10's of labeled examples.

4.3 Annotation costs and Active Selection

In the following sections we evaluate how well we can learn to predict the difficulty of segmenting images using image features and the impact of using the predicted cost when making an active selection.

4.3.1 Annotation cost prediction

First, we isolate how well we can learn to predict the difficulty of segmenting images based on image features. To train our cost function, we gather data with Amazon's Mechanical Turk. Users are required to completely segment

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**

Fig. 19 Accuracy of our cost function in predicting "easy" vs. "hard", both for user-specific and user-independent classifiers.

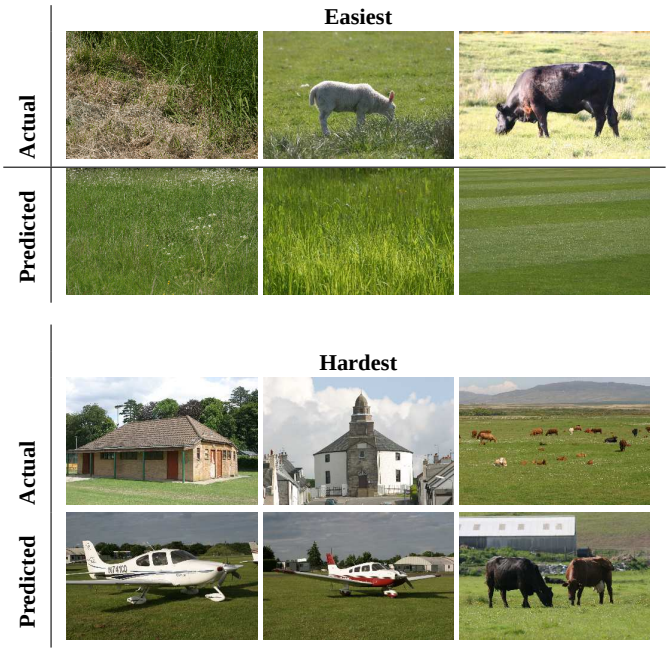


Fig. 20 The easiest and hardest images to annotate based on actual users' timing data (top), and the predictions of our cost function on novel images (bottom).

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; each user was allowed to label up to 240 images. However, no user completed all 240 images. The fact that most users skipped certain images (Figure 19, column: Number of images) 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 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 (0.61, 0.33 respectively), with the rest near zero. Figure 19 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 20 shows examples that were easiest and hardest to segment, as measured by the ground truth actual time taken for at least eight 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 21 (left) 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.

In order to verify that we were not simply learning a category-based level of effort, we looked at the actual and predicted times split across different classes. Figure 21 (right) shows a plot of the actual and predicted times broken across the different scene settings in the MSRC dataset. The x-axis shows the most dominant foreground class label in that particular scene layout. This figure shows that every class/scene layout contains images with varying difficulty in terms of the annotation effort required by users. While some categories have more variation than others (cow vs car) there is no direct connection between the image class and the time taken to provide annotations. The plot also shows that for most of the examples our cost predictor provides fairly accurate predictions of the annotation costs.

4.3.2 Active selection with a learned cost function

Thus far we have fixed the costs assigned per annotation type; now we show the impact of using the predicted cost while making active choices. We train a binary multi-instance classifier for each MSRC category using image labels on $\frac{4}{5}$-th of the data per class, in five 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 22 shows representative (good and bad) learning curves, with accuracy measured by the AUROC value. For

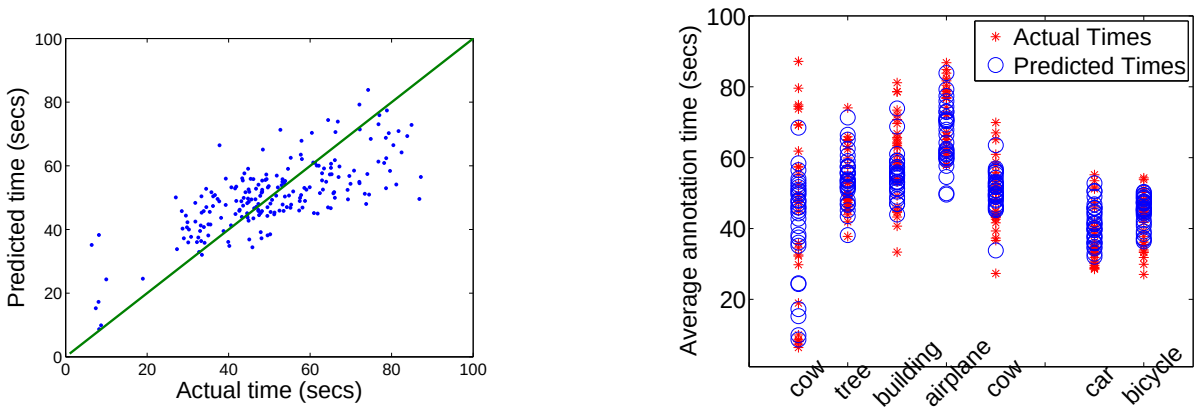


Fig. 21 **Left:** Scatter-plot of the actual time taken by users to segment an image vs. the value predicted by our cost function, for the 240 images in the MSRC v1. The predicted and actual times are highly correlated, implying that our cost predictor has learned how difficult an image is to segment using only low-level image features. **Right:** The actual and predicted times split across the different categories of images in the MSRC dataset. The plot shows that most classes have images with varying difficulties, and assures that the difficulty measure we have learned is not class-specific.

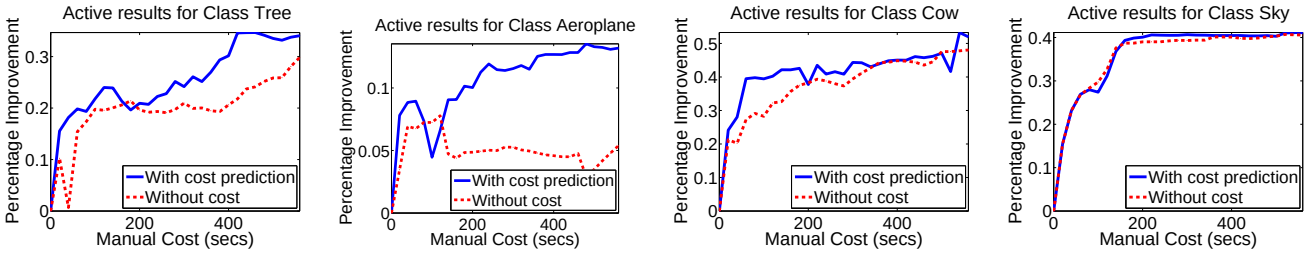


Fig. 22 Representative learning curves when using active selection with the learned cost predictor, as compared to a baseline that makes active selections using a flat cost value. For classes like Tree, Cow, and Airplane (shown here), the cost prediction produces more improvement per unit cost, while for a few like Sky there is no significant difference—most likely because the images within the class are fairly consistent and equally informative and easy to label.

% acc 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

Table 2 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.

Tree, Cow, and Airplane, using the predicted cost leads to 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.

Table 2 shows the cost required to improve the base classifier to different levels of accuracy. The fourth column shows the relative time savings our cost prediction enables over a cost-blind active learner that uses the same 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.

5 Conclusions

Our approach addresses a new problem: how to actively choose not only which instance to label, but also what type of image annotation to acquire in a cost-effective way. Our method is general enough to accept other types of annotations or classifiers, as long as the cost and risk functions can be appropriately defined. We have shown that compared to traditional active learning which restricts supervision to yes/no questions, a richer means of providing supervision and a method to effectively select supervision based on both information gain and cost to the supervisor is better-suited for building classifiers with minimal human intervention.

There are several directions of future work for our research. The foremost is to reduce the computational complexity of the active selection criterion. With our implementation of the incremental SVM technique of Cauwenberghs and Poggio (2000) 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. This corresponds to about 15 minutes to choose which annotation to request when the dataset contains ~ 100 bags (images) for $\sim$20 classes. Once an annotation is se-

lected it takes less than 0.1 secs to retrain the classifier. The most expensive step in selecting an annotation is the Gibbs sampling procedure coupled with the need to update a large number of classifiers in the one-vs-one setting. 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.

Currently, we are exploring the problem of cost-sensitive batch selection, where the goal is to actively choose a set of examples for labeling at once, while ensuring that the total annotation request costs less than a given budget (Vijayanarasimhan et al, 2010).

Additionally, if we wanted to use our method with the intention of targeting specific annotators who have variable capabilities and speeds depending on image content, we could build user-specific cost functions, i.e., a separate SVM for each. Then, we could extend the VOI to choose not only what annotation type and image looks most promising, but also which user ought to be responsible for annotating it.

Allowing further levels of supervision, such as scene layout, contextual cues or part labels, would enable us to improve the way in which human supervisors can interact with computer vision systems. Generative models could be more suited to integrate such disparate cues. Extending the approach to generative models is another direction of research we are planning to pursue.

Finally, while we have concentrated mostly in the domain of object recognition, the problem of comparing different types annotations in a unified framework is potentially applicable to several other domains both in vision and machine learning such as video annotation, tracking, or document classification.

Acknowledgements. Many thanks to Alex Sorokin for helping us arrange the Mechanical Turk data collection. This research was supported in part by NSF CAREER IIS-0747356, Microsoft Research, DARPA VIRAT, NSF EIA-0303609, and the Henry Luce Foundation.

References

von Ahn L, Dabbish L (2004) Labeling Images with a Computer Game. In: CHI

Bach FR, Lanckriet GRG, Jordan MI (2004) Fast Kernel Learning using Sequential Minimal Optimization. Tech. Rep. UCB/CSD-04-1307

Baldridge J, Osborne M (2008) Active Learning and Logarithmic Opinion Pools for Hpsg Parse Selection. Nat Lang Eng 14(2):191–222

Bart E, Ullman S (2005) Cross-Generalization: Learning Novel Classes from a Single Example by Feature Replacement. In: CVPR

Bunescu RC, Mooney RJ (2007) Multiple Instance Learning for Sparse Positive Bags. In: ICML

Cauwenberghs G, Poggio T (2000) Incremental and Decremental Support Vector Machine Learning. In: NIPS

Collins B, Deng J, Li K, Fei-Fei L (2008) Towards Scalable Dataset Construction: An Active Learning Approach. In: ECCV

Dietterich TG, Lathrop RH, Lozano-Perez T (1997) Solving the Multiple Instance Problem with Axis-Parallel Rectangles. Artif Intell 89(1-2):31–71, DOI http://dx.doi.org/10.1016/S0004-3702(96)00034-3

Fei-Fei L, Fergus R, Perona P (2003) A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories. In: ICCV

Fergus R, Fei-Fei L, Perona P, Zisserman A (2005) Learning Object Categories from Google's Image Search. In: ICCV

Gartner T, Flach P, Kowalczyk A, Smola A (2002) Multi-Instance Kernels. In: ICML

Greiner R, Grove AJ, Roth D (2002) Learning Cost-Sensitive Active Classifiers. Artif Intell 139(2):137–174

Haertel R, Ringger E, Seppi K, Carroll J, McClanahan P (2008) Assessing the Costs of Sampling Methods in Active Learning for Annotation. In: Proceedings of Workshop on Parsing German

Kapoor A, Grauman K, Urtasun R, Darrell T (2007a) Active Learning with Gaussian Processes for Object Categorization. In: ICCV

Kapoor A, Horvitz E, Basu S (2007b) Selective Supervision: Guiding Supervised Learning with Decision-Theoretic Active Learning. In: IJCAI

Kwok JT, Cheung P (2007) Marginalized Multi-Instance Kernels. In: IJCAI

Lanckriet GRG, Cristianini N, Bartlett P, Ghaoui LE, Jordan MI (2004) Learning the Kernel Matrix with Semidefinite Programming. J Mach Learn Res 5:27–72

Lee Y, Grauman K (2008) Foreground Focus: Finding Meaningful Features in Unlabeled Images. In: BMVC

Li L, Wang G, Fei-Fei L (2007) Optimol: Automatic Online Picture Collection via Incremental Model Learning. In: CVPR

Maron O, Ratan AL (1998) Multiple-Instance Learning for Natural Scene Classification. In: ICML

Platt J (1999) Advances in Large Margin Classifiers, MIT Press, chap Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods

Qi G, Hua X, Rui Y, Tang J, Zhang H (2008) Two-Dimensional Active Learning for Image Classification. In: CVPR

Quelhas P, Monay F, Odobez JM, Gatica-Perez D, Tuytelaars T, VanGool L (2005) Modeling Scenes with Local Descriptors and Latent Aspects. In: ICCV

Russell B, Torralba A, Murphy K, Freeman W (2005) Labelme: a Database and Web-Based Tool for Image Annotation. Tech. rep., MIT

Settles B, Craven M, Ray S (2008) Multiple-Instance Active Learning. In: NIPS

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

Sivic J, Russell B, Efros A, Zisserman A, Freeman W (2005) Discovering Object Categories in Image Collections. In: ICCV

Sorokin A, Forsyth D (2008.) Utility Data Annotation with Amazon Mechanical Turk. In: CVPR Workshops

Verbeek J, Triggs B (2007) Region Classification with Markov Field Aspect Models. In: CVPR

Vijayanarasimhan S, Grauman K (2008a) Keywords to Visual Categories: Multiple-Instance Learning for Weakly Supervised Object Categorization. In: CVPR

Vijayanarasimhan S, Grauman K (2008b) Multi-Level Active Prediction of Useful Image Annotations for Recognition. In: NIPS

Vijayanarasimhan S, Grauman K (2009) What's It Going to Cost You?: Predicting Effort vs. Informativeness for Multi-Label Image Annotations. In: CVPR

Vijayanarasimhan S, Jain P, Grauman K (2010) Far-Sighted Active Learning on a Budget for Image and Video Recognition. In: CVPR

Weber M, Welling M, Perona P (2000) Unsupervised Learning of Models for Recognition. In: ECCV

Winn J, Criminisi A, Minka T (2005) Object Categorization by Learned Universal Visual Dictionary. In: ICCV

Wu TF, Lin CJ, Weng RC (2004) Probability Estimates for Multi-Class Classification by Pairwise Coupling. JMLR

Yan R, Yang J, Hauptmann A (2003) Automatically Labeling Video Data using Multi-Class Active Learning. In: ICCV

Yang C, Lozano-Perez T (2000) Image Database Retrieval with Multiple-Instance Learning Techniques. In: ICDE

Zha ZJ, Hua XS, Mei T, Wang J, Qi GJ, Wang Z (2008) Joint Multi-Label Multi-Instance Learning for Image Classification. In: CVPR

Zhou ZH, Zhang ML (2006) Multi-Instance Multi-Label Learning with Application to Scene Classification. In: NIPS