Entropy-Based Active Learning for Object Recognition

Alex Holub, Pietro Perona
Caltech
1200 E. California Blvd. Pasadena, CA 91106
holub@vision.caltech.edu, perona@vision.caltech.edu

Michael C. Burl
Jet Propulsion Laboratory, Caltech
1200 E. California Blvd. Pasadena, CA 91106
Michael.C.Burl@jpl.nasa.gov

Abstract

Most methods for learning object categories require large amounts of labeled training data. However, obtaining such data can be a difficult and time-consuming endeavor. We have developed a novel, entropy-based "active learning" approach which makes significant progress towards this problem. The main idea is to sequentially acquire labeled data by presenting an oracle (the user) with unlabeled images that will be particularly informative when labeled. Active learning adaptively prioritizes the order *in which the training examples are acquired, which, as shown by our experiments, can significantly reduce the overall* number *of training examples required to reach near-optimal performance. At first glance this may seem counter-intuitive: how can the algorithm know whether a group of unlabeled images will be informative, when, by definition, there is no label directly associated with any of the images? Our approach is based on choosing an image to label that maximizes the expected amount of information we gain about the set of unlabeled images. The technique is demonstrated in several contexts, including improving the efficiency of web image-search queries and open-world visual learning by an autonomous agent. Experiments on a large set of 140 visual object categories taken directly from text-based web image searches show that our technique can provide large improvements (up to 10x reduction in the number of training examples needed) over baseline techniques.*

1. Introduction

There are many situations in computer vision where the cost of obtaining labeled data is extremely high. Consider the problem of obtaining sufficient training data to build recognizers for thousands of image categories; clearly, one needs to be as efficient as possible when confronted with such a large number of categories and images. By intelligently choosing the subset of images to be labeled, we may be able to dramatically reduce the *number* of images needed for the labeled training set. Work on support vector machines [12], relevance vector machines [9], and other sparse classifiers has shown that not all examples are created equal, as these classifiers express their solutions (decision surfaces) in terms of a small subset of the full set of examples.

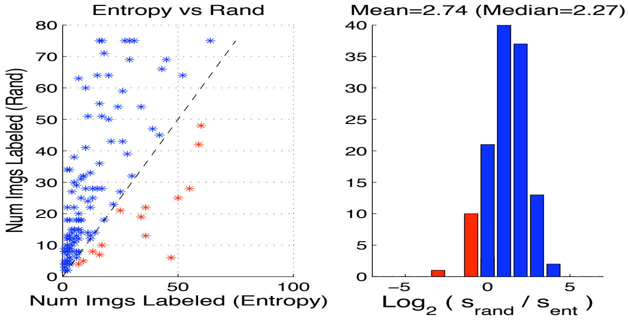

Figure 2. Comparison of Minimum Expected Entropy (MEE) active learning against passive learning (random sampling) over 137 categories of image search data (see an example category in Figure 1). (Left) Scatter plot showing the number of labeling rounds to reach 85% of asymptotic maximum performance for MEE (x-axis) versus random sampling (y-axis). Points above the diagonal indicate that MEE reaches near-optimal performance with fewer labeling rounds than passive learning. Each point represents a different object category and is the result of averaging over 50 experiments for the category. All experiments used an unlabeled pool of 250 images. (Right) Histogram of the $\log_2$ speedup (multiplicative improvement) of MEE versus passive learning. We indicate the mean and median increase in performance in title, i.e., a mean of 4, indicates that on average active learning reached target performance $4\times$ faster than random learning.

Figure 1 shows that similar issues arise when performing text-based searches for a particular object class. A basic search may return a high percentage of images that do not match the target concept. If we could refine these searches by acquiring user input, we could drastically increase the precision of the returned results. However, since user time is precious, it is critical that we attempt to squeeze as much information as possible from a minimum amount of feedback.

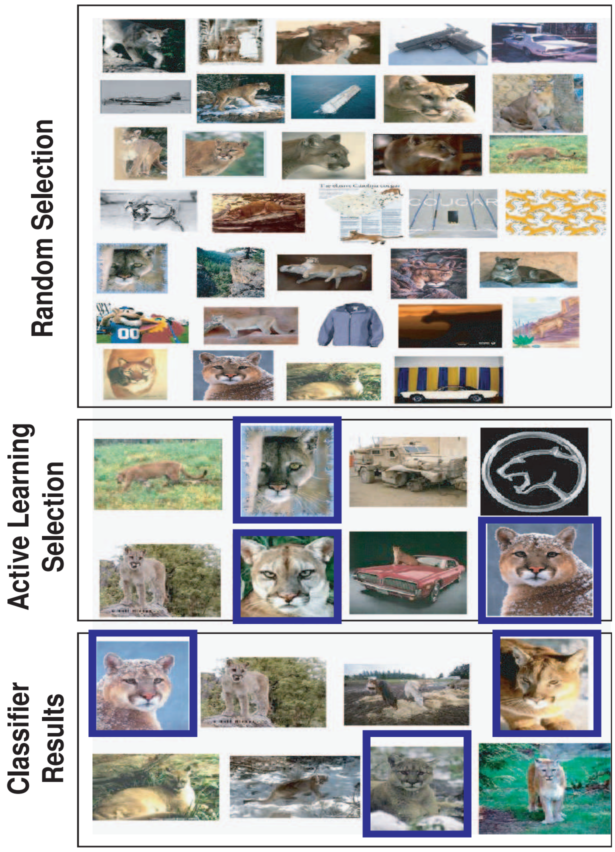

Figure 1. Web image search for category 'Cougar'. The user is allowed to label images to refine the image search query. (Top) Images the user needs to label in passive learning (randomly choosing images for the user to label) in order to achieve 82% of maximum performance. (Middle) Images sequentially selected by active learning for the user to label. The blue boxes indicate images the user has marked as 'Good'. Note that the user is required to rate over 4× fewer images when active learning is used compared to passive learning. In this example, the user prefers images that show the head of a cougar. (Bottom) The top 8 returns of the resulting classifier trained using the active learning images. Blue boxes indicate images which are 'Good' according to the user. Figure 2 shows similar performance gains for 137 image search categories.

Finally, consider an autonomous agent traversing a world and encountering new object classes. The agent is allowed to query an oracle (e.g., a human) regarding information found in the world, but, as the oracle's time is valuable, the number of queries must be kept to a minimum (e.g., consider a Mars rover, which must consume precious resources and time to query human 'oracles' back on Earth).

In this paper we employ active learning to more quickly and efficiently learn visual object categories. In general active learning paradigms have 4 key components: (1) a set of labeled training examples, (2) a set of unlabeled examples for which labels can be obtained at some cost, (3) an oracle (e.g., a human) that can provide correct labels, and (4) a methodology for deciding which unlabeled examples to request labels for given the current state of knowledge. Typically, this process occurs iteratively so that unlabeled examples are selected and then labeled by the oracle. Given the new information from the oracle, additional unlabeled examples are selected for labeling. Colloquially, we refer to the image selected for labeling at each iteration as the "Most Informative Unlabeled Point" (the MIUP). This formulation of active learning is similar to that described in [7].

The main issue then is determining how to select the next image to label given what is currently known. Many heuristics have been developed; one of the most common is to choose examples which are the "most confused" with respect to the current classifier being used. For instance a confused point might be the point which lies closest to the decision surface separating two classes. Tong and Koller [11] develop this idea for Support Vector Machines (SVMs) by looking at the closest point to the current separating hyperplane. This idea has been further developed for image retrieval experiments [10]. Seung et al. [8] take a different approach to selecting the most-confused point by generating numerous viable classifiers based on the known labels and choosing the point to label as the one that is most-confused by these classifiers. The authors of [2] compare yet another "most confused" point approach and apply it to image retrieval experiments.

Relevance feedback, a related method, has been studied for content-based image retrieval (CBIR) systems since the mid-1990's. Many of these techniques focus on learning similarity measures between images or on weighting the importance of low-level features such as shape, color, and texture in defining the user's target concept. See [13] for a review.

We take a different approach to active learning in the hopes of improving performance and improving the flexibility of the active learning approach. Given the multitude of image classifiers available, we would like our active learning approach to be agnostic w.r.t. the underlying image classifier being used. We suggest to choose the MIUP which results in acquiring the most information about the unlabeled images (similar to [6], but in a classification rather than regression setting), or, expressed another way, which minimizes the expected uncertainty of the unlabeled set of images. Our algorithm, in the same spirit as [8], generates numerous viable classifiers in order to identify the MIUP and is well-defined for any underlying classifier being used (in this paper we demonstrate the technique on SVMs, Nearest Neighbor, and Kernel Nearest Neighbor classifiers).

A recent paper [3] uses active learning and Pyramid Match Kernels to improve object category recognition. They employ a Gaussian Process (GP) model to place a prior probability on the spatial correlation of the underlying labels. Their algorithm then estimates a posterior probability distribution over the label of any unlabeled point given the currently labeled points. They use a variation of Most Confused Point (MCP) taking into account the posterior variance to choose the MIUP.

In Section 2 we describe our algorithm. In Section 3, we describe the data-sets used and show that our technique provides substantial speedup over competing methods on a large set of 137 image categories. We also consider the important question of deciding automatically when enough labeled data has been acquired. Finally in Section 4, we conclude and discuss implications of this work. Two technical appendices are included, which provide a brief overview of several alternative active learning approaches that were tested and the Lazebnik Spatial Pyramid Match Kernel [4] (used as the underlying classification method in our experiments).

2. Active Learning

We formalize our discussion of active learning as follows. Suppose we have a set of N images with each image belonging to one of L possible classes. Initially we assume that the class labels for all images are unknown. Active learning begins by choosing one or more of the N images; these images are presented to an oracle that provides the correct class label(s). In subsequent rounds, the active learning algorithm considers both the the currently labeled images and the remaining unlabeled images and chooses additional images from the unlabeled set that would be particularly informative if their labels were known.

Let $\mathcal{U}^{(t)}$ be the pool of unlabeled images at the start of round t and let $\mathcal{L}^{(t)}$ be the corresponding pool of labeled images. Initially, we have $\mathcal{U}^{(0)}$ containing all N images and $\mathcal{L}^{(0)} = \emptyset$. For simplicity of notation, we will assume that one unknown image is to be chosen in each round and assigned a label, although see Section 2.4 for a discussion of the multi-return case.

To admit both deterministic and random algorithms, we suppose that an active learning algorithm outputs an $(M \times 1)$ vector $\mathbf{w}$ that specifies a probability distribution over the images in the unlabeled pool, where M is the number of unlabeled images available in the current round. A deterministic algorithm simply sets all the elements of $\mathbf{w}$ to zero, except for one element which is set to 1. (This element is then guaranteed to be picked.) *Random sampling* (equivalent to passive learning) sets $\mathbf{w}$ to $1/M \cdot \underline{\mathbf{1}}$, where $\underline{\mathbf{1}}$ is an $(M \times 1)$ vector of ones. Given $\mathbf{w}$, the oracle chooses an image according to this distribution and returns its label. This process leads to new labeled and unlabeled sets for the next round.

$$\mathcal{L}^{(t+1)} = \mathcal{L}^{(t)} \cup \{\mathbf{x}^{(t)}, y^{(t)}\} \tag{1}$$
$$\mathcal{U}^{(t+1)} = \mathcal{U}^{(t)} \setminus \mathbf{x}^{(t)} \tag{2}$$

where $\mathbf{x}^{(t)} \in \mathcal{U}^{(t)}$ is the example chosen in round t and $y^{(t)}$ is its label assigned by the oracle.

2.1. Minimum Expected Entropy

The usual goal with active learning is to learn, as quickly as possible, a decision function $g(\cdot)$ that accurately assigns class labels y to test images $\mathbf{x}$. However, given the uncertainties involved (even the form of the underlying class-conditional probability distributions is unknown), it is difficult to directly optimize this criterion. Instead, we have developed a novel active learning approach that attempts to sequentially minimize the expected entropy (uncertainty) of the labels for the unlabeled images given the current labeled set.

Let $\mathcal{H}(\cdot)$ represent the entropy of a set of images. In round t, we want to choose the image that produces the maximum reduction in entropy (equivalently, maximum gain in information) once its label is known.

$$\mathbf{x}^{(t)} = \arg\max_{\mathbf{x}} \mathcal{H}\left(\mathcal{U}^{(t)}|\mathcal{L}^{(t)}\right) - \mathcal{H}\left(\mathcal{U}^{(t+1)}|\mathcal{L}^{(t+1)}\right) \tag{3}$$

Since only the second term depends[1] on $\mathbf{x}$), we can instead solve the following *minimization*:

$$\mathbf{x}^{(t)} = \arg\min_{\mathbf{x}} \mathcal{H}\left(\mathcal{U}^{(t+1)}|\mathcal{L}^{(t+1)}\right) \tag{4}$$

There is a problem with our formulation so far. In both Equations 3 and 4, we have an entropy conditional on $\mathcal{L}^{(t+1)}$, which presumes we know the label that the oracle will assign to $\mathbf{x}$. Since this label information is unknown before we consult the oracle, $\mathcal{H}\left(\mathcal{U}^{(t+1)}|\mathcal{L}^{(t+1)}\right)$ cannot be calculated. To resolve this issue, we instead compute an entropy conditional on each possible result the oracle might give for the label of $\mathbf{x}$. We then average these conditional entropies weighted by the probability that $\mathbf{x}$ takes on a particular label to generate an *expected entropy*:

$$\overline{\mathcal{H}}_{\mathbf{x}} = \sum_{j=1}^{L} P(Y=j|\mathcal{L}^{(t)}) \cdot \mathcal{H}\left(\mathcal{U}^{(t+1)}|\mathcal{L}^{(t)} \cup \{\mathbf{x}, j\}\right) \tag{5}$$

where Y is a random variable representing the label of $\mathbf{x}$. The Minimum Expected Entropy (MEE) algorithm chooses the image that results in the minimum value for $\overline{\mathcal{H}}_{\mathbf{x}}$.

$$\mathbf{x}^{(t)} = \arg\min_{\mathbf{x}} \overline{\mathcal{H}}_{\mathbf{x}} \qquad \text{(MEE)} \tag{6}$$

The main difficulty in implementing MEE is to estimate $\mathcal{H}(\mathcal{U}|\mathcal{L})$. (The superscripts that indicate the epoch number have been dropped to simplify the notation.) This quantity is the *joint entropy* over the random variables Y_k representing the labels of the unlabeled images conditional on $\mathcal{L}$.

[1] The dependence is implicit; $\mathbf{x}$ is the new image from $\mathcal{U}^{(t)}$ to be labeled.

The joint entropy, of course, depends on the full joint probability distribution over the vector of Y variables, which is difficult to estimate. Therefore, we make use of the subadditivity property of entropy: $\mathcal{H}\left(Y_1, Y_2, \ldots, Y_M | \mathcal{L}\right) \le \sum_{k=1}^{M} \mathcal{H}\left(Y_k | \mathcal{L}\right)$ to replace the joint entropy by a sum over the individual (marginal) entropies; this new quantity serves as an upper bound for the joint entropy. (The bound is tight if the Y_k's are independent.)

To estimate $\mathcal{H}(Y_k|\mathcal{L})$, we simply need to know the probability distribution over the possible label values that Y_k can take, then the entropy is given by:

$$\mathcal{H}(Y_k|\mathcal{L}) = -\sum_{l=1}^{L} P(Y_k = l|\mathcal{L}) \cdot \log_2 P(Y_k = l|\mathcal{L}) \quad (7)$$

Estimation of the label probabilities is discussed in the next subsection. Algorithm 2.1 provides a pseudocode summarization of the Minimum Expected Entropy approach.

```
for each round t do
  for each unlabeled image $\mathbf{x}_i \in \mathcal{U}^{(t)}$ do
    for each possible class label $j \in \{1, \ldots, L\}$ do
      Estimate $P\left(Y_i = j|\mathcal{L}^{(t)}\right)$
      for each unlabeled image $\mathbf{x}_k \in \left(\mathcal{U}^{(t)} \backslash \mathbf{x}_i\right)$ do
        for each possible class label $l \in \{1, \ldots, L\}$ do
          Estimate $P\left(Y_k = l|\mathcal{L}^{(t)} \cup \{\mathbf{x}_i, j\}\right)$
        end for
      end for
      Calculate conditional entropy $\mathcal{H}_j$
    end for
    Combine conditional entropies $\mathcal{H}_j$ for $j = 1, \ldots, L$
    into an expected entropy $\overline{\mathcal{H}}_{\mathbf{x}_i}$
  end for
  Set w to $\delta_{i_*}$ where $\mathbf{x}_{i_*}$ yields lowest expected entropy.
end for
```

2.2. Look-ahead Estimate of Class Probabilities

Here we consider how to estimate class probabilities for the unlabeled images given a set of labeled images $\mathcal{L}$ when we have classifiers, such as kernel nearest neighbor or SVM, that only return hard class decisions[2]. The key idea is to use a one step look-ahead scheme to construct a committee of classifiers. The predictions of the committee are then used to derive the desired label probabilities. The look-ahead step considers each of the M currently unlabeled images in $\mathcal{U}$ and each possible value for its class label. Let $\{\mathbf{x}_m, n\}$ be a look-ahead image and its hypothesized label. Next we construct a classifier from $\mathcal{L} \cup \{\mathbf{x}_m,\, n\}$. Repeating this process for each unlabeled image and each possible value for its label yields $M \cdot L$ classifiers, which we apply to each of the images in $\mathcal{U}$.

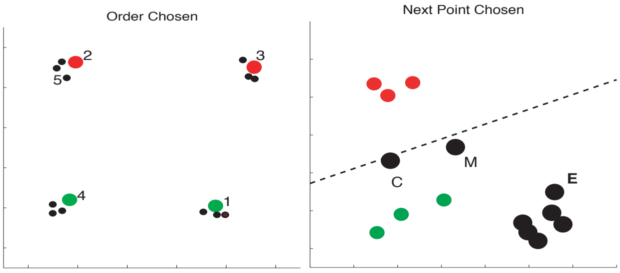

Figure 3. (Left) Illustration of minimum expected entropy (MEE) active learning for a set of $N = 20$ points from $L = 2$ classes. The numbers show the order in which MEE presents points to the oracle with the colored circles showing the resulting label. With its first four queries, MEE visits each of the "clusters" providing for the quickest reduction in the uncertainty of the labels of the other points. This experiment was run using a Nearest Neighbor classifier. (Right) Comparison of different active learning approaches. The red and green points are currently labeled, while the black points are unlabeled. The dotted line shows the SVM hyperplane found with the current set of labeled points. The next query to the oracle is shown for three different active learning approaches: (E) MEE, (C) closest to the current hyperplane as in [11], and (M) most confused point based on lookahead estimates for the label probabilities. Observe that MEE chooses a point that sits next to many other unlabeled points *and* is in a relatively unexplored region of space. Note that our MEE framework allows us to use *any* underlying classifier: Nearest Neighbor classifiers are used in the left figure while SVMs are used in the right.

The classifier results can be collected into a block-structured matrix $\mathbf{B}$ consisting of L blocks by L blocks with each block being an $(M \times M)$ matrix. (As usual, L is the number of class labels and M is the number of currently unlabeled images.) The (l, n) block contains an indicator matrix of 0's and 1's. The (k, m) position within a block is a 1 if the classifier trained with $\mathcal{L} \cup \{\mathbf{x}_m,\, n\}$ says that example $\mathbf{x}_k$ belongs in class l. We can then use B to write:

$$\underline{\mathbf{P}}(Y_k = l) \quad = \quad \frac{1}{M} \cdot \mathbf{B} \cdot \underline{\mathbf{P}}(Y_m = n) \qquad (8)$$

(The notation is such that the probability vectors $\underline{\mathbf{P}}(\cdot)$ on the LHS and RHS are stacked up in "label-major" order.) The RHS probability vector acts like a prior probability on the labels of points. The LHS is analogous to a posterior, re-estimated after we see the predictions of the committee of look-ahead classifiers. This equation can be understood from either a histogram viewpoint or an expectation viewpoint. From the histogram viewpoint, we are accumulating the probability that a classifier selected randomly from the committee says "1" to the event $(Y_k = l)$. From the expectation viewpoint, we are computing over the entire committee an expectation for the binary-valued classifier confidence in the event $(Y_k = l)$. (This duality exists because

[2] Although the SVM hyperplane distance can be used to construct pseudo-probabilities, this approach cannot be applied to other types of classifiers.

the expectation of a binary-valued random variable $E[X]$ is the same as $P(X = 1)$.)

From Equation 8, we can obtain an estimate for the class probabilities by taking $P(Y_m = n) = 1/L \cdot \underline{\mathbf{1}}$ on the RHS and multiplying by $1/M \cdot \mathbf{B}$. In principle, this process can be iterated to refine the probabilities. In the limit, the probabilities satisfy the solution of a fixed point problem $\mathbf{p} = 1/M \cdot \mathbf{Bp}$. Solving this equation amounts to finding the eigenvector of $\mathbf{B}/M$ corresponding to eigenvalue 1. (Since the $(1 \times M \cdot L)$ vector $\underline{\mathbf{1}}^T$ is a left eigenvector of $\mathbf{B}/M$ with (left) eigenvalue 1, we know that there exists a right eigenvector of $\mathbf{B}/M$ with eigenvalue 1.) Although the iterative and fixed point approaches are elegant, in practice we have found from a limited set of experiments that a single iteration of Equation 8 with uniform probabilities on the RHS yields better results. We are still studying this issue, but believe the main cause is that Equation 8 does not adequately reflect the possibility that look-ahead classifiers trained from a finite amount of data are simply wrong. Iterating causes the estimation procedure to develop unwarranted certainty about the class labels.

2.3. Computational Cost

The MEE algorithm described above, although intuitively appealing, is somewhat expensive computationally. In particular, the algorithm is $O(L^3N^3)$ with N the number of unlabeled images and L the number of classes. A typical run on 2Ghz Pentium using $N = 250$ and $L = 2$ and a combination of Matlab and C code takes about 30 minutes.

We consider how the benefits of active learning change with the size of the pool of unlabeled images. Figure 4 addresses this point and shows that increasing the pool tends to constantly increase the performance of Entropy-based active learning over random sampling.

Given the benefit of increasing pool size, we consider methods of reducing the computational cost. One possibility is to use only a fixed number of images M when calculating the expected entropy for a particular images, such that we reduce the $O(L^3N^3)$ to $O(L^3N\tilde{N}^2)$ for some constant $\tilde{N} < N$. In particular, we randomly sample a set M unlabeled images from the entire pool of unlabeled images to compute the entropy with. Figure 5 illustrates the effects of this sub-sampling of the unlabeled pool and shows that performance tends to drop off fast when subsets are used. There are many other methods for increasing speed and we leave these open as topics for further research. However we note that we were able to easily run experiments using 250 unlabeled images and a non-optimized code.

2.4. Multi-Return Active Learning

So far, we have viewed active learning as presenting a single image (or a probability distribution $\mathbf{w}$ for selecting a single image) to the oracle for each round of labeling. In practice, it will often be preferable to return a *set* of images rather than the single most informative image. Consider web image search. The interaction with the user would be cumbersome if they were asked to label only a single image at a time; instead, it would be preferable for the user to label a *set* of images. There are also technical reasons for returning multiple images at once as illustrated in Figure 6. Analogous to greedy forward feature selection algorithms, single-return active learning picks three images that do not cover the space as well as if the three images were picked at once as a unit.

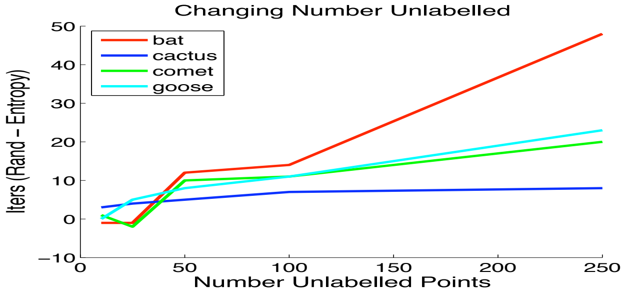

Figure 4. Results from four categories showing how the difference in time (iterations) required for random and MEE to reach 85% of maximum performance varies as the size of the unlabeled pool is increased. The relative advantage of active learning is clearly more pronounced when the unlabeled pool is larger.

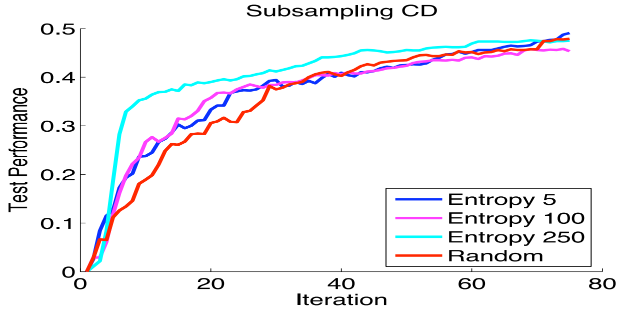

Figure 5. Here we consider the effects of only using a subset of the available images to compute the entropy. X-axis: the number of images labeled. Y-axis: performance on a separate test set of data. There are a total of 250 unlabeled images available, and each line represents using a subset of those points. Red represents randomly choosing images. Note that performance falls off quickly as less images are used to compute entropy, i.e. best performance results from using all 250 images in the unlabeled pool. Results shown are using the category 'CD', but are typical of other categories as well.

Multi-return active learning using the minimum expected entropy principle requires only a minor modification to Algorithm 2.1. In particular, the loop over unlabeled images $(\mathbf{x}_i)$ is replaced by a loop over subsets of unlabeled images of size s. For each subset, we consider the L^s possible assignments of labels to the elements in the subset and

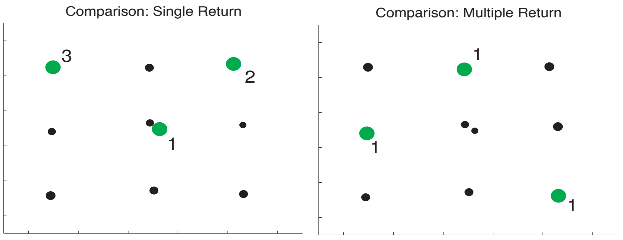

Figure 6. (Left) Single-return active learning applied three times. (Right) Multi-return active learning applied once with subsets of size 3. Clearly, the multi-return approach is able to generate a more optimal covering of the space.

# Categories	# Images	Good	Bad
137	30277	93 (82/134)	178 (56/402)

Table 1. Table detailing the large collection of images obtained from web image searches. The total number of categories, the total number of images, the mean and min / max number of images in each of the categories labeled as 'Good' and the same for 'Bad'.

compute an expected entropy as before. The subset that results in the lowest expected entropy is then presented to the oracle for labeling. Exhaustively considering all subsets of size s in each active learning round is clearly only feasible for small s; however, given that the information value of a proposed subset can be easily evaluated (using the expected entropy), other heuristics can be incorporated to focus consideration onto a smaller number of promising subsets.

3. Experiments

There are two key sets of experiments we performed. The first is inspired by Figure 1 and involves increasing the precision of web image searches. This is essentially a two-category task, discriminating images that match the target concept from those that do not. The second set of experiments considers many (up to 10) categories and is inspired by an autonomous agent exploring a world.

3.1. Web Image Searching

Consider again Figure 1 in which the user typed 'Cougar' into an image search engine. The idea is that the user must label a set of images in order to refine the search as most of the returned images do not contain the category of interest. In this case active learning can provide a drastic increase in speed by choosing the MIUP. In this section we explore experiments designed to mimic just such situations.

3.1.1 Image Search Dataset

Our goal in this set of experiments was to mimic as closely as possible a real image search on the web. We collected images returned from actual text-based web image searches with Google and PicSearch. In order to obtain comprehensive statistics we collected images using 137 keywords. We next asked 3 sorters to label the images as one of three classes: 'Good', 'Ok', 'Bad'. The 'Good' images contain images of the class of interest while the 'Bad' images do not contain the object of interest [3]. We removed all duplicate images using software which first extracts features using the Lowe Difference of Gaussian detector and SIFT descriptors [5] and then compares these sets of features across all images in the category. If there are more than 100 good matches between two images, the images are considered to be identical and one is removed. For our experiments we only used images from the 'Good' and 'Bad' categories. This is the largest data-set of its type to our knowledge. Table 1 gives some statistics on the data-set we collected. The full set of category names are too numerous to list here, but are provided in the Supplementary Materials.

3.1.2 Results

Our experiments were conducted as follows. For each category we combined the 'Good' and 'Bad' images into a single large pool. From this pool we randomly selected a set of 75 testing images. The rest of the images were used as the pool of unlabeled data for active learning. We then followed Algorithm 2.1 and iteratively chose images to label using MEE active learning. We also considered alternative approaches including: (1) random sampling (passive learning) choosing a image, (2) choosing the most confused image, and (2) choosing the unlabeled image with highest kernel density (see Appendix 1 for an overview of these alternative-methods. In all cases, kernel nearest neighbor using the Spatial Pyramid Match Kernel of Lazebnik [4] was used as the classifier in our experiments (see Appendix 2 for an overview or [4] for full details).

How do we quantify performance? In these experiments we are interested in the precision for the top 25 closest images. In other words what percentage of the 25 closest returned images are in the 'Good' class? Let p_{max} be the maximum possible performance on the test set (this occurs when all images in the initial unlabeled pool get labeled). Now consider the number of images, $s_i, i \in (0, 1, 2, 3)$ which need to be labeled to achieve 85% of p_{max} where i indexes over the various active learning methods. Results showing the performance of MEE active learning versus the alternative methods are presented in Figures 2 and 7. Note that MEE significantly outperforms all of these competing methods. In fact we reach 85% of p_{max} up to $10\times$ faster using MEE to pick the MIUP when compared to random and perform on average close to $3\times$ better than random on these data-sets.

[3] We will make both the positive images and negative images publicly available pending acceptance.

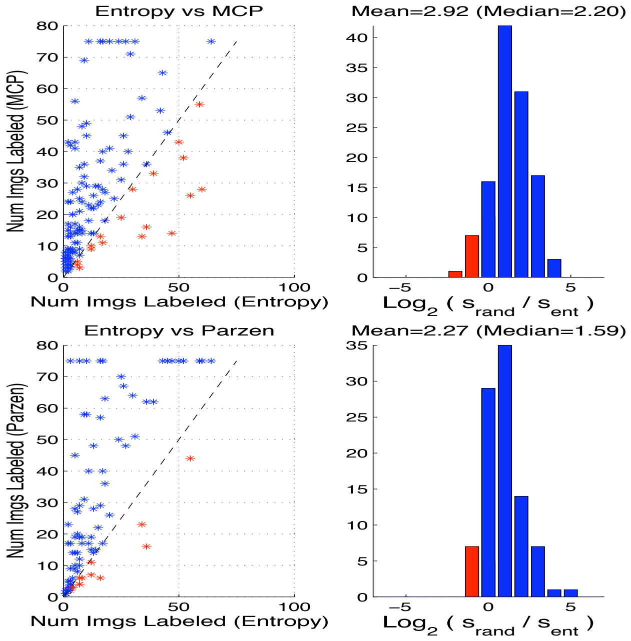

Figure 7. Comparison of MEE active learning with: (top) choosing the Most Confused Point and (bottom) the Maximum Unlabeled Density (sampling the highest density of unlabeled points.

3.2. Exploration Agent

The next set of experiments looks at multiple classes. We motivate this experiment by considering an agent traveling through a real or virtual world (for instance a mobile robot exploring the environment or a web-crawler). This agent will be confronted with a wealth of visual information. With minimal supervision can the agent discover and learn to recognize multiple categories of objects? Given that there is a considerable cost associated with obtaining a label for any particular image (e.g., the agent must ask a human observer whose time is precious), for which images should the agent request labels?

3.2.1 Open-World Learning Experiments

In these experiments, the unlabeled pool has examples from many object categories. Initially our agent has no knowledge of the world and assumes there is only a single object class. The agent chooses informative images via active learning and asks an oracle to label these images; the oracle returns the true label of the unknown image. As new classes are encountered the agent updates its knowledge of the number of classes which exist in its world (L) increases. Here, we use a slightly different criteria from the Image Search experiments to assess performance. Consider that in this scenario the agent is seeking to build the best classifiers for visual categorization, and thus we consider the classification performance on a separate set of test data. Our experiments were conducted as follows. First we selected the Good examples from L different categories. From these we randomly choose a set of N images to form $\mathcal{U}$, the pool of unlabeled examples. The rest of the images are used as a test set with which to evaluate the performance of our algorithm.

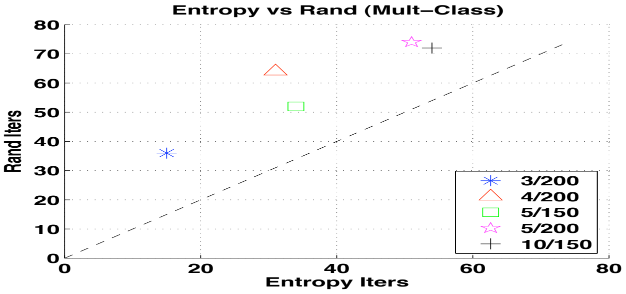

Figure 8. Similar to Fig. 2 but multi-class experiments. Each point represents an experiment indicated by the legend. The legend indicates two numbers: the first is the number of classes used and the second is the size of the pool of unlabeled data. Each point is the average over 25 iterations, where each iteration involves choosing a random set of categories and training data for each category.

3.2.2 When Have Enough Images Been Labeled?

A natural question which arises is: when has the agent learned enough about the environment? Or, when should the agent stop querying the oracle? Our MEE framework allows us to estimate the entropy $\mathcal{H}^{(t)}$ after each active learning iteration and hence the amount of information gained after each active learning iteration can be approximated by: $\mathcal{I}^{(t)} = \mathcal{H}^{(t)} - \mathcal{H}^{(t-1)}$. In Figure 9 we consider the relationship between $\mathcal{I}^{(t)}$ and the performance gains on the test set. A strong relationship exists between the change in information and the change in performance of the system. We can use MEE to estimate when we have acquired sufficient information about the unlabeled images.

4. Discussion

We have developed a novel "active learning" algorithm that enables hundreds of complex object categories to be recognized with a minimal amount of labeled training data. Our approach uses a principled, information-theoretic criteria to select the most informative images to be labeled. The technique is well-defined for any underlying classifier (kernel nearest neighbor, SVM, etc.), extends naturally to multi-class and multi-return settings, and can automatically determine when enough labeled training data has been acquired to insure near-maximal recognition performance. Against passive learning and a variety of alternative active learning approaches, our method consistently achieves near-maximal performance with one-half to one-third the

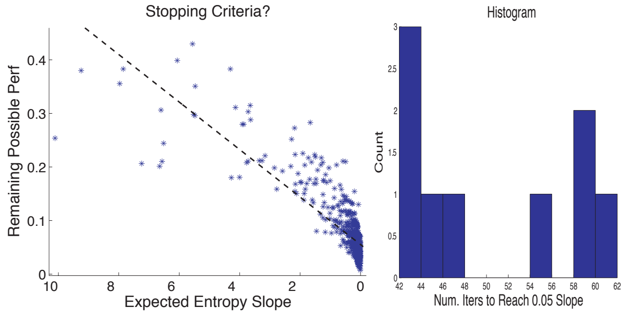

Figure 9. Can we use MEE to determine when to stop learning? (Left) Scatter plot. X-axis: the slope of the MEE at a particular iteration t. Y-axis: the remaining performance gain possible at the same iteration t (the difference between the current performance and the maximum possible performance). Dotted black line is a regression over all the points. A steep entropy slope correlates with large potential increases in performance, indicating we should keep learning. A shallow entropy slope (near zero) correlates with little potential for performance increase indicating we should stop sampling. (Right) Histogram for different 5 class 200 unlabeled image experiments. The x-axis is the time taken to reach a particular slope value less than .05. It takes different experiments longer to reach a shallow slope, and from the left figure, a shallow slope indicates very little potential for performance increase, we can label substantially fewer images for some experiments using MEE as a stopping criteria.

number of training and in some cases the improvement is 10x or more.

Appendix 1: Other Active Learning Methods

We compare the MEE approach to two other approaches: the Most-Confused-Point (MCP) and Maximum Unlabeled Density (MUD). MCP follows the spirit of [11], choosing the image which is most confused between the different classes. To calculate the most confused image, we follow the paradigm of Sec 2.2 to estimate class probabilities for each image. The MCP based on these probability estimates is selected. The MUD technique estimates a probability density $p(\mathbf{x}|\mathcal{U})$ over the unlabeled points using Parzen Window kernel density estimation. The unlabeled point with the maximum probability density is selected for labeling, i.e., $\mathbf{x}^{(t)} = \arg\max_j \sum_i \frac{1}{N} K(\mathbf{x}_j, \mathbf{x}_i)$ where i and j are indices for unlabeled images. The MUD technique is comparable to clustering the unlabeled data and choosing a point near the center of the most prominent cluster. The drawback is that we do not distinguish between high densities of unlabeled points and high densities of unlabeled points with uncertain labels. Figure 7 compares these approaches.

Appendix 2: Pyramid Match Kernel

Spatial Pyramid Matching [4] was used as it performs well on data-sets similar to those in this paper [1] and is fast. For each image, we extract a set of SIFT features [5]. 10,000 features are chosen at random from a training set of images in order to form a vocabulary of $M = 200$ words, and the vocabulary is used to map each subsequent feature to one of the 200 words. The image is split into a 4×4 grid and the number of times each of the 200 features is found in each of the 16 bins is counted. The matching kernel is computed using the above set of $4 \times 4 \times M$ histograms. *Matching* means finding the number of common elements in any two bins. If the counts in two bins are n_1 and n_2 the match is $\min(n_1, n_2)$. Matching is computed with spatial information and appearance.

Acknowledgements

This research has been carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration (NASA). Funding provided under the Applied Information Systems Research Program. A.H. and P.P. were supported by N00014-06-1-0734.

References

[1] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report UCB/CSD-04-1366, California Institute of Technology, 2007. 8

[2] F. Jing, M. Li, H.-J. Zhang, and B. Zhang. Entropy-based active learning with support vector machines for content-based image retrieval. *IEEE Int. Conf. on Mult and Expo*, 2004. 2

[3] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. *ICCV*, 2007. 2

[4] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, 2006. CVPR. 3, 6, 8

[5] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 6, 8

[6] D. MacKay. Information-based objective functions for active data selection, 1992. Neural Computation. 2

[7] M. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification, 1998. ICML. 2

[8] H. Seung, , M. Opper, and H. Sompolinsky. Query by committee, 1992. Proceedings of the Fifth Workshop on Computational Learning Theory. 2

[9] M. Tipping. The relevance vector machine, 2000. NIPS. 1

[10] S. Tong and E. Chang. Support vector machine active learning for image retrieval, 2001. Proceedings of the ninth ACM international conference on Multimedia. 2

[11] S. Tong and D. Koller. Support vector machine active learning with applications to text classification, 2000. ICML. 2, 4, 8

[12] V. Vapnik. The nature of statistical learning theory, 1995. 1

[13] X. Zhou and T. Huang. Relevance feedback in image retrieval: A comprehensive review. *Mult. Systems*, 2003. 2