Scalable Active Learning for Multiclass Image Classification

Ajay J. Joshi, *Member*, *IEEE*, Fatih Porikli, *Senior Member*, *IEEE*, and Nikolaos P. Papanikolopoulos, *Fellow*, *IEEE*

Abstract—Machine learning techniques for computer vision applications like object recognition, scene classification, etc., require a large number of training samples for satisfactory performance. Especially when classification is to be performed over many categories, providing enough training samples for each category is infeasible. This paper describes new ideas in multiclass active learning to deal with the training bottleneck, making it easier to train large multiclass image classification systems. First, we propose a new interaction modality for training which requires only yes-no type binary feedback instead of a precise category label. The modality is especially powerful in the presence of hundreds of categories. For the proposed modality, we develop a Value-of-Information (VOI) algorithm that chooses informative queries while also considering user annotation cost. Second, we propose an active selection measure that works with many categories and is extremely fast to compute. This measure is employed to perform a fast seed search before computing VOI, resulting in an algorithm that scales *linearly* with dataset size. Third, we use locality sensitive hashing to provide a very fast approximation to active learning, which gives *sublinear time scaling*, allowing application to very large datasets. The approximation provides up to two orders of magnitude speedups with little loss in accuracy. Thorough empirical evaluation of classification accuracy, noise sensitivity, imbalanced data, and computational performance on a diverse set of image datasets demonstrates the strengths of the proposed algorithms.

Index Terms—Active learning, scalable machine learning, multiclass classification, object recognition

1 Introduction

Real-world classification applications such as object recognition and classification typically require large amounts of annotated training data due to the tremendous amount of variation in image appearance. Considering the variety and scale of images on the web, training satisfactory classifiers is increasingly difficult using traditional supervised learning techniques. At the same time, in most image classification problems, we typically have a large number of unlabeled data. Intelligently exploiting the large amounts of data is a challenging problem. To this end, there has been recent interest in *active learning*, wherein classifiers are trained interactively—the system queries the user for annotations on "informative samples" instead of accepting annotation passively. Previous work on binary classification [39] and, more recently, even multiclass classification [17], [20], [23], [43] has shown that such an active learning approach can reduce the amount of training data required compared to supervised passive learning.

Even though multiclass active learning methods successfully reduce the amount of training data required, they can be labor intensive from a user interaction standpoint for the following reasons:

1. For each unlabeled image queried for annotation, the user has to sift through many categories to input the precise one. Especially for images, providing input in this form can be difficult and sometimes impossible when a huge (or unknown) number of categories are present.
2. The time and effort required increase with an increase in the number of categories.
3. The interaction is prone to mistakes in annotation.
4. It is not easily amenable to distributed annotation as all users need to be consistent in labeling. The distributed annotation aspect is of increasing importance in exploiting distributed labeling resources such as Amazon Mechanical Turk [1], [37].

Apart from the above, current active learning algorithms are computationally intensive, which limits their applicability to datasets of hundreds or a few thousand samples. At the same time, image datasets are ever increasing in their size and the image variety—it is not uncommon to have tens of thousands of image classes [7], [40]. In order to design systems that are practical at larger scales, it is essential to allow easier modes of annotation and interaction for the user, along with algorithms that are scalable. Motivated by this, our contributions in this paper are the following:

- We develop a multiclass active learning setup that requires only binary user feedback (yes/no). Our system generalizes interaction since it can also accept precise category annotation as in the traditional setting, if available for any images. We

- *A.J. Joshi is with Google, Inc., 1600 Amphitheatre Pkwy., Mountain View, CA 94043. E-mail: ajay@cs.umn.edu.*
- *F. Porikli is with Mitsubishi Electric Research Labs, 201 Broadway 8th Floor, Cambridge, MA 02139. E-mail: fatih@merl.com.*
- *N.P. Papanikolopoulos is with the Department of Computer Science and Engineering, University of Minnesota, Twin Cities, 4-192 Keller Hall, 200 Union Street SE, Minneapolis, MN 55455. E-mail: npapas@cs.umn.edu.*

Manuscript received 14 Dec. 2010; revised 27 Aug. 2011; accepted 15 Dec. 2011; published online 9 Jan. 2012.
Recommended for acceptance by J. Winn.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2010-12-0953.
Digital Object Identifier no. 10.1109/TPAMI.2012.21.

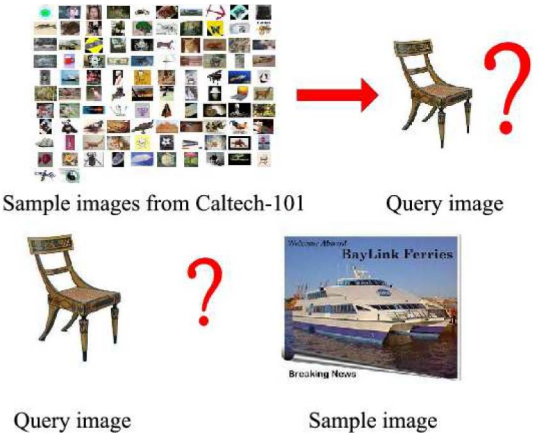


Fig. 1. Top row: Sample interaction in traditional multiclass active learning approaches. The user needs to input a category name/number for the query image from a large dataset possibly consisting of hundreds of categories. Bottom row: The binary interaction model we propose. The user only needs to say whether or not the query image and the sample image belong to the same category.

propose a Value-of-Information (VOI)-based active selection framework for the binary interaction modality.

- We propose an efficient measure to compute uncertainty (uncertainty sampling) of examples for query selection. Unlike previous work, the selection measure works *directly in the multiclass setting*, instead of actively selecting samples from various binary classifiers separately.
- We propose extremely efficient approximations to active learning that scale *sublinearly* with dataset size. Scaling is of utmost importance toward using active learning for current applications, which are at a substantially larger scale than what most algorithms are designed for.
- Unlike most previous methods, the proposed system is designed to handle and incorporate unseen categories as it learns (i.e., we do not assume a training set containing samples from all classes to begin with). This aspect is particularly important in real systems, where it is unlikely to have training examples from all categories at the outset.
- The proposed system is empirically shown to handle many frequent issues that plague real data: class population imbalance owing to largely varying number of examples across categories, label noise occurring due to human training errors, or noisy acquisition processes.

Due to the ease of interaction of the proposed system, easy scalability, allowing the incorporation of unseen categories, and dealing with noise and imbalance, we believe this paper demonstrably shows the effectiveness of active learning for training very large-scale image classification systems.

1.1 Ease of Interaction

In order to quantitatively compare the two interaction modalities, we conducted experiments on 20 users with 50-class and 100-class data, obtained from the Caltech-101 object categories dataset [12]. Each user was asked to interact with two modalities, as shown in Fig. 1: 1) giving category labels (out of a given set of labels) to randomly queried images, as is typically used for training, and 2) giving yes/no responses to two images based on whether they came from the same class. We measured interaction time and the number of errors made in both modalities by each user, along with an overall satisfaction score from 1 through 5, indicating the ease of interaction experienced (1 being the easiest). Table 1 summarizes the results.

TABLE 1
Comparing the Two Interaction Modalities

Modality	Response time (s)	% errors	Satisfaction
BF – 50 classes	1.6 ($\pm$0.2)	0.80	1.2
MCF – 50 classes	11.7 ($\pm$3.1)	12.7	4.1
BF – 100 classes	1.7 ($\pm$0.2)	0.82	1.1
MCF – 100 classes	28.8 ($\pm$5.3)	14.3	4.9

First, it can be seen that binary feedback (BF) requires far less user time than giving multiclass feedback (MCF). Although BF in principle also provides lesser information than MCF, we demonstrate in our experiments that the BF interaction model still achieves superior classification accuracy than MCF with the same expenditure of user time. Second, as seen in the table, MCF has much more noise associated—users make many more errors when sifting through potential categories and finding the correct one. In contrast, BF is much cleaner since it is much easier to simply look at two images and determine whether they belong to the same class or not. Third, the interaction time and annotation errors in MCF increase with the number of categories. This is expected, as annotation requires browsing over all possible classes.

In contrast, in the BF model there is no observed increase in user time with increasing number of categories. This aspect is particularly appealing, as the main objective is to scale well to larger problems with potentially thousands of classes. Four, as seen from the satisfaction scores, users are much more satisfied with the overall interaction in BF since it does not need browsing through many images and can be done quickly. Apart from the above advantages, distributed annotation across many trainers is easily possible in the BF model. Also, it is straightforward to allow exploration of the data when new categories continuously appear (as opposed to a setting often used previously wherein the initial training set is created by including examples from all classes [15]), or when notions of categories change with time. In summary, binary feedback provides an extremely appealing interaction model for large problems with many classes.

2 Related Work

Many problems in computer vision suffer from the fact that they require substantial amounts of training data for performing accurate classification. As such, active learning has received increasing interest in the computer vision community. In the following, we review relevant work on object recognition and active learning.

Tong and Chang [38] propose active learning for SVM in a relevance feedback framework for image retrieval. Their approach relies on the margins for unlabeled examples for binary classification. Tong and Koller [39] use an active

learning method to minimize the version space[1] at each iteration. However, both these approaches target binary classification.

Gaussian processes (GP) have been used for object categorization by Kapoor et al. [23]. They demonstrate an active learning approach through uncertainty estimation based on GP regression, which requires $\mathcal{O}(N^3)$ computations, cubic in the number of training examples. They use one-versus-all SVM formulation for multiclass classification, and select one example per classifier at each iteration of active learning.

Holub et al. [17] propose an entropy (EP)-based active learning method for object recognition. Their method selects examples from the active pool, whose addition to the training set minimizes the *expected entropy* of the system. On the other hand, our VOI method computes the expected improvement in classification accuracy, while also attempting to minimize the expected user annotation cost. The entropy-based approach proposed in [17] requires $\mathcal{O}(k^3N^3)$ computations, where N is the number of examples in the active pool and k is the number of classes. Qi et al. [34] demonstrate a multilabel classification method that employs active selection along two dimensions—examples and their labels. Label correlations are exploited for selecting the examples and labels to query the user.

Kapoor et al. [24] propose a VOI type method for semi-supervised learning which is similar to the one proposed here. Our approach, however, proposes a simpler binary interaction model for multiclass problems, along with an associated efficient means to compute VOI on the binary model.

For handling multiple image selection at each iteration, Hoi et al. [16] introduced batch mode active learning with SVMs. Since their method is targeted toward image retrieval, the primary classification task is binary—to determine whether an image belongs to the class of the query image. Active learning with uncertainty sampling has been demonstrated by Li and Sethi [28], in which they use conditional error as a metric of uncertainty and work with binary classification.

For a comprehensive survey on various algorithms and applications of active learning, see [36]. Although there has been a lot of work on reducing the number of training examples for classification, the interaction and computational complexity of active learning has been more or less overlooked, especially for classification tasks involving large number of categories. As mentioned previously, addressing this is the primary contribution of the paper.

2.1 Learning Setup

Fig. 2 shows a block schematic of the proposed active learning setup. The active pool consists of a large number of unlabeled images from which the active learning algorithm can select images to query the user. The training set consists of images for which category labels are known and can be used for training the classifier. Throughout the paper, we use Support Vector Machines (SVM) as the underlying classification algorithm since it provides state-of-the-art performance on the datasets used for evaluation. For the multiclass case, one-versus-one SVM (classifiers trained for each pair of classes) are used.

1. Version space is the subset consisting of all hypotheses that are consistent with the training data [30].

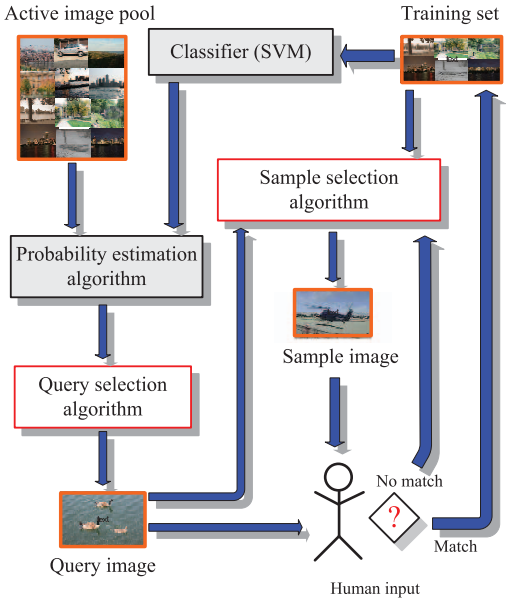


Fig. 2. Block schematic of the active learning setting. Our focus in this paper is on the query and sample selection algorithms—depicted in white boxes with red borders (see text for details).

In the traditional multiclass active learning setting, an unlabeled image (query image) needs to be selected for user annotation. In our case, however, since user input is only binary, we also require an image from a known category to show the user for comparison. Selecting this image from the training set is a new aspect of active selection that our framework requires. We refer to this comparison image from a known category as the "sample image." We focus on query and sample selection algorithms in this paper—denoted by white boxes with red borders in Fig. 2.

Our approach for query as well as sample selection is probabilistic, i.e., based on the current training set, class membership probability estimates are obtained for the images in the active pool. We use Platt's method [29], [33] to estimate binary probabilities based on the SVM margins, combined with pairwise coupling [42] with one-versus-one SVM for multiclass probability estimation on the unlabeled images. Probability estimation details are given in Section 4.1.

In Fig. 2, the query selection algorithm selects a query image from the active pool using the estimated class membership probabilities. Based on the estimated membership probabilities for the query image, the sample selection algorithm selects a sample image from the current training set. The query-sample pair is shown to the user for feedback. If a "match" response is obtained, indicating that the query and sample images belong to the same category, the query image is added to the current training set along with its category label. If a "no-match" response is obtained, the sample selection algorithm is again invoked to ask for a different sample image.

This process goes on until either the label for the query image is obtained (with a "match" response) or until the query image does not match any of the categories in the

training set. In the latter case, a new category label is initiated and assigned to the query image.[2] Through such a mechanism, the learning process can be started with very few training images initially chosen at random (seed set). As the process continues, the active selection algorithm requires far fewer queries than random selection to achieve similar classification rate on a separate test set. Note that the system is also able to exploit feedback in terms of precise category annotation (as in the typical setting), if available. Binary feedback, however, generalizes the applicability and allows learning in new unknown environments for exploration.

Binary input has been employed previously in the context of clustering data by asking the user for pairwise must-link and cannot-link constraints [3]. This approach can be adapted to the active learning framework by choosing even the sample images from unlabeled data and performing a (unsupervised) clustering step before user annotation. However, in our observation, such an approach was prone to noise due to unsupervised clustering, which can lead to an entire cluster of incorrectly labeled training data. Noise reduction in the preclustering approach is an interesting future work direction. On the other hand, in this paper (and the preliminary version [22]), we demonstrate empirically that the setup we employ is robust to labeling noise.

3 The Active Learning Method

There are two parts to binary feedback active learning: 1) to select a query image from the active pool, and 2) to select a sample image from a known category to be shown to the user along with the query image.

3.1 Query Selection

The goal here is to query informative images, i.e., images that are likely to lead to an improvement in future classification accuracy. We use the Value of Information framework [24], [25], [41] employed in decision theory for query selection in this paper. The broad idea is to select examples based on an objective function that combines the misclassification risk and the cost of user annotation. Consider a risk matrix $M \in \mathbb{R}^{k\times k}$ for a k-class problem. The entry M_{ij} in the matrix indicates the risk associated with misclassifying an image having true label i as belonging to class j. Correct classification incurs no risk and hence the diagonal of M is zero, $M_{ii} = 0$, $\forall i$.

Denote the estimated class membership distribution for an unlabeled image x as $\mathbf{p}_x = \{p_x^1, \ldots, p_x^k\}$. Note that since the true class membership distribution for x is unknown, the actual misclassification risk cannot be computed—we instead find the *expected* misclassification risk for x as

$$\mathcal{R}_{\mathcal{L}}^{\{x\}} = \sum_{i=1}^{k}\sum_{j=1}^{k} M_{ij}\cdot\left(p_x^i|\mathcal{L}\right)\cdot\left(p_x^j|\mathcal{L}\right), \tag{1}$$

where $\mathcal{L}$ is the set of labeled examples based on which the probabilities are estimated. Consider that the test set $\mathcal{T}$ consists of N images $x_1, \ldots, x_N$. The total expected risk over the test set (normalized by size) is

$$\mathcal{R}_{\mathcal{L}} = \frac{1}{|\mathcal{T}|}\sum_{x\in\mathcal{T}}\sum_{i=1}^{k}\sum_{j=1}^{k} M_{ij}\cdot\left(p_x^i|\mathcal{L}\right)\cdot\left(p_x^j|\mathcal{L}\right). \tag{2}$$

Note that the above expression requires that the test set be available while computing the total risk. Typically the test set is not available beforehand, and we can use the images in the active pool $\mathcal{A}$ for computing the expected risk. Indeed, most work on classification uses surrogates to estimate the misclassification risk in the absence of the test set. In many scenarios, the entire available set of unlabeled images is used as the active pool and is typically very large; thus an estimate of risk on the active pool is fairly reliable.

Now, if $y \in \mathcal{A}$ is added to the labeled training set by acquiring its label from the user, the expected reduction in risk on the active pool can be computed as

$$\begin{aligned}\mathcal{R}_{\mathcal{L}} - \mathcal{R}_{\mathcal{L}'} = &\frac{1}{|\mathcal{A}|}\sum_{x\in\mathcal{A}}\sum_{i=1}^{k}\sum_{j=1}^{k} M_{ij}\cdot\left(p_x^i|\mathcal{L}\right)\cdot\left(p_x^j|\mathcal{L}\right)\\ &-\frac{1}{|\mathcal{A}'|}\sum_{x\in\mathcal{A}'}\sum_{i=1}^{k}\sum_{j=1}^{k} M_{ij}\cdot\left(p_x^i|\mathcal{L}'\right)\cdot\left(p_x^j|\mathcal{L}'\right),\end{aligned} \tag{3}$$

where $\mathcal{L}' = \mathcal{L}\cup\{y\}$, and $\mathcal{A}' = \mathcal{A}\setminus\{y\}$. The above expression captures the *value* of querying y and adding it to the labeled set. However, we also need to consider the *cost* associated with obtaining feedback from the user for y. Assume that the cost of obtaining user annotation on y is given by $\mathcal{C}(y)$. In our framework, we wish to actively choose the image that reduces the cost incurred while maximizing the reduction in misclassification risk. Assuming risk reduction and annotation cost are measured in the same units, the joint objective that represents the VOI for a query y is

$$V(y) = \mathcal{R}_{\mathcal{L}} - \mathcal{R}_{\mathcal{L}'} - \mathcal{C}(y). \tag{4}$$

The term $\mathcal{R}_{\mathcal{L}}$ in the above equation is independent of y, the example to be selected for query. Therefore, active selection for maximizing VOI can be expressed as a minimization

$$y^* = \operatorname*{argmin}_{y\in\mathcal{A}} \mathcal{R}_{\mathcal{L}'} + \mathcal{C}(y). \tag{5}$$

Note that the above framework can utilize any notions of risk and annotation cost that are specific to the domain. For instance, we can capture the fact that misclassifying examples belonging to certain classes can be more expensive than others. Such a notion could be extremely useful for classifying medical images so as to determine whether they contain a potentially dangerous tumor. Misclassifying a "clean" image as having a tumor only incurs the cost of the doctor verifying the classification. However, misclassifying a "tumor image" as clean could be potentially fatal in a large dataset wherein the doctor cannot manually look at all the data. In such scenarios, the different misclassification risks could be suitably encoded in the matrix M.

As in most work on active learning, our evaluation is based on classification accuracy. As such, we employ equal misclassification cost so that $M_{ij} = 1, \text{ for } i \neq j$.

3.2 Sample Selection

Given a query image, the sample selection algorithm should select sample images so as to minimize the number of responses the user has to provide. In our framework, the

2. Initiating a new category can require many user responses when many classes are present—we later discuss how to overcome this through a fast new class initialization step along with cluster merging.

sample images belong to a known category; the problem of selecting a sample image then reduces to the problem of *finding a likely category for the query image* from which a representative image can be chosen as the sample image. When presented with a query image and a sample image, note that a "match" response from the user actually gives us the category label of the query image itself! A "no match" response does not provide much information. Suppose that the dataset consists of 100 categories. A "no match" response from the user to a certain query-sample image pair still leaves 99 potential categories to which the query image can belong. Based on this understanding, the goal of selecting a sample image is to maximize the likelihood of a "match" response from the user.

Selecting a sample image (category) can be accomplished by again using the estimated class membership probabilities for the selected query image. For notational simplicity, assume that the query image distribution $\{p_1, \ldots, p_k\}$ is in sorted order such that $p_1 \geq p_2 \geq \cdots \geq p_k$. The algorithm proceeds as follows: Select a representative sample image from class 1 and obtain user response. As long as a "no match" response is obtained for class $i-1$, select a sample image from class i to present the user. This is continued until a "match" response is obtained. Through such a scheme, sample images from the more likely categories are selected earlier in the process in an attempt to minimize the number of user responses required.

3.2.1 Annotation Cost

In the binary feedback setting, our experiments indicated that it is reasonable to assume that each binary comparison requires a constant cost (time) for annotation. Thus, for each query image, the cost incurred to obtain the class label is equal to the number of binary comparisons required. Since this number is unknown, we compute its expectation based on the estimated class membership distribution instead. If the distribution is assumed to be in sorted order as above, the expected number of user responses to get a "match" response is

$$\mathcal{C}(x) = p_1^x + \sum_{j=2}^{k} \left(1 - p_1^x\right) \ldots \left(1 - p_{j-1}^x\right) \cdot p_j^x \cdot j, \qquad (6)$$

which is also the user annotation cost. We can scale the misclassification risk (by scaling M) with the real-world cost incurred to find the true risk, which is in the same units as annotation cost. Here, we choose the true risk as the *expected number of misclassifications* in the active pool, and compute it by scaling M with the active pool size. Along with our choice of $\mathcal{C}(x)$, this amounts to equating the cost of each binary input from the user to every misclassification, i.e., we can trade one binary input from the user for correctly classifying one unlabeled image.

3.3 Stopping Criterion

The above VOI-based objective function leads to an appealing stopping criterion—we can stop whenever the maximum expected VOI for any unlabeled image is **negative**, i.e., $\operatorname{argmax}_{x \in \mathcal{A}} V(x) < 0$. With our defined notions of risk and cost, negative values of VOI indicate that a single binary input from the user is not expected to reduce the number of misclassifications by even one; hence querying is not worth the information obtained. It should be noted that different notions of real-world risk and annotation cost could be employed instead if specific domain knowledge is available. The selection and stopping criteria directly capture the particular quantities used.

3.4 Initiating New Classes

Many active learning methods make the restrictive assumption that the initial training set contains examples from all categories [15]. This assumption is unrealistic for most real problems since the user has to explicitly construct a training set with all classes, defeating our goal of reducing supervision. Also, if a system is expected to operate over long periods of time, handling new classes is essential. Thus, we start with small seed sets, and allow dynamic addition of new classes. In the sample selection method described above, the user is queried by showing sample images until a "match" response is obtained. However, if the query image belongs to a category that is not present in the current training set, many queries will be needed to initiate a new class.

Instead, we initiate a new class when a fixed small number (say 5) of "no-match" responses are obtained. With good category models, the expected distributions correctly capture the categories of unlabeled images—hence, "no-match" responses to the few most likely classes often indicates the presence of a previously unseen category. However, it may happen that the unlabeled image belongs to a category present in the training data. In such cases, creating a new class and assigning it to the unlabeled image results in overclustering. This is dealt with by agglomerative clustering (cluster merging), following the min-max cut algorithm [8], along with user input.

The basic idea in agglomerative clustering is to iteratively merge two clusters that have the highest similarity (linkage value) $l(C_i, C_j)$. For min-max clustering, the linkage function is given by $l(C_i, C_j) = s(C_i, C_j)/(s(C_i, C_i)s(C_j, C_j))$, where s indicates a cluster similarity score: $s(C_i, C_j) = \sum_{x \in C_i} \sum_{y \in C_j} K(x, y)$. Here, K is the kernel function that captures similarity between two objects x and y (the same kernel function is also used for classification with SVM).

In our algorithm, we evaluate cluster linkage values after each iteration of user feedback. If the maximum linkage value (indicating cluster overlap) is for clusters C_i and C_j and is above a threshold of 0.5, we query the user by showing two images from C_i and C_j. A "match" response results in merging of the two clusters. Note that our setting is much simpler than the unsupervised clustering setting since we **have user feedback available**. As such, the method is relatively insensitive to the particular threshold used, and lesser noise is encountered. Also, note that we do not need to compute the linkage values from scratch at each iteration—only a simple incremental computation is required. In summary, new classes are initiated quickly and erroneous ones are corrected by cluster merging with little user feedback.

3.5 Computational Considerations

The computational complexity of each query iteration in our algorithm (Fig. 3) is $\mathcal{O}(N^2k^3)$, with an active pool of size N and

Input: Labeled set $\mathcal{L}$, active pool $\mathcal{A}$, cost matrix M

$\mathcal{L}^0 := \mathcal{L}$; $\mathcal{A}^0 := \mathcal{A}$
for round $r = 0$ **to** $n - 1$ **do**
  foreach image $x_i \in \mathcal{A}^{(r)}$ **do**
    for class $y_i = 1$ **to** k **do**
      Train multi-class classifier with $\mathcal{L}^{(r)} \cup \{x_i, y_i\}$
      Estimate class membership probabilities for images in the active pool $\mathcal{A}^{(r)}$
      Compute risk on the active pool $R^{(x_i,y_i)}$
    end
    Compute expected risk ($\mathcal{L}'^r = \mathcal{L}^r \cup \{x_i\}$) $\mathcal{R}_{\mathcal{L}'^r} = \sum_l P(y_i = l) \cdot R^{(x_i,l)}$
    Compute expected annotation cost $\mathcal{C}(x_i)$
  end
  Find image $x^* = \text{argmin}_{x_i \in \mathcal{A}^{(r)}} \mathcal{R}_{\mathcal{L}'^r} + \mathcal{C}(x_i)$
  Find $V(x^*)$ using Eqn. (4)
  if $V(x^*) > 0$ **then**
    Query user with query image x^* and likely sample images until true label k^* is obtained
    Set $\mathcal{L}^{(r+1)} := \mathcal{L}^{(r)} \cup \{x^*, k^*\}$; and $\mathcal{A}^{(r+1)} := \mathcal{A}^{(r)} \setminus \{x^*\}$
  else return $\mathcal{L}^{(n)} = \mathcal{L}^{(r)}$
end

Output: The new labeled set $\mathcal{L}^{(n)}$

Fig. 3. Multiclass active learning with binary feedback.

k classes. Although it works well for small problems, the cost can be impractical at larger scales. In the following, we propose a new uncertainty measure for active selection in multiclass scenarios that allows extremely fast computation. The measure can be seen as one way to define margins in the multiclass case. We will then use the proposed selection measure to seed the search by restricting the number of examples over which VOI has to be computed.

4 Multiclass Active Selection Measure

Our approach follows the idea of uncertainty sampling [4], [13], wherein examples on which the current classifier is uncertain are selected to query the user. Distance from the hyperplane for margin-based classifiers has been used as a notion of uncertainty in previous work. However, this does not easily extend to multiclass classification due to the presence of multiple hyperplanes. We use a different notion of uncertainty that is easily applicable to a large number of classes. The uncertainty can be obtained from the class membership probability estimates for the unlabeled examples as output by the multiclass classifier. In the case of a probabilistic model, these values are directly available. For other classifiers, such as SVM, we need to first estimate class membership probabilities of the unlabeled examples. In the following, we outline our approach for estimating the probability values for multiclass SVM. However, such an approach for estimating probabilities can be used with many other nonprobabilistic classification techniques also.

4.1 Probability Estimation

In order to obtain class membership probability estimates for unlabeled examples in the active pool, we follow the approach proposed by Lin et al. [29], which is a modified version of Platt's method to extract probabilistic outputs from SVM [33].

The basic idea is to approximate the class probability using a sigmoid function. Suppose that $x_i \in \mathbb{R}^n$ are the feature vectors, $y_i \in \{-1, 1\}$ are their corresponding labels, and $f(x)$ is the decision function of the SVM which can be used to find the class prediction by thresholding. The conditional probability of class membership $P(y = 1|x)$ can be approximated using

$$p(y = 1|x) = \frac{1}{1 + \exp(Af(x) + B)}, \tag{7}$$

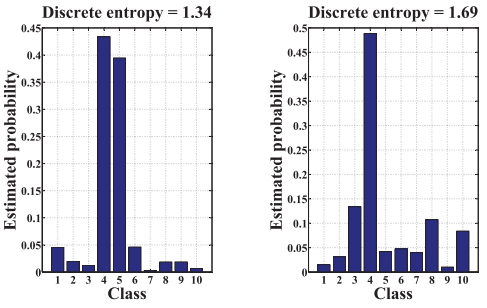
where A and B are parameters to be estimated. Maximum likelihood estimation is used to solve for the parameters. In order to generate probability estimates from the binary classifiers described above, pairwise coupling [42] was used. Please see [21] for details on the probability estimation method. We used the toolbox LIBSVM [5] that implements the SVM for classification and probability estimation in the multiclass problem.

4.2 Pairwise Classification

As shown above, we use pairwise SVM for classification. Consequently, $\mathcal{O}(k^2)$ classifiers are required for a k class problem. Even though training in the pairwise classifiers setting appears to be computationally inefficient compared to the one-versus-all setting, which requires k classifiers, pairwise classifiers can in fact be equally efficient or, occasionally, even more so than the one-versus all setting for the following reasons. In the one-versus-all setting, we need to train k classifiers with N (training set sample size) data samples each, assuming no sampling approximations. On the other hand, assuming relatively equal distribution of samples across all the classes, each classifier in the pairwise setting is trained with about $2N/k$ samples. Further noting that SVM training scales approximately quadratically with the training set size, the pairwise setting often results in *faster* training on the entire dataset. Along with faster training, pairwise classifiers result in better prediction in our experiments. Computational efficiency of pairwise classification has also been demonstrated previously in [18], and its superior classification performance was noted by Duan and Keerthi [9].

4.3 Entropy as Uncertainty

Each labeled training example belongs to a certain class, denoted by $y \in \{1, \ldots, k\}$. However, we do not know true class labels for examples in the active pool. For each unlabeled example, we can consider the class membership variable to be a random variable denoted by Y. We have a distribution $\mathbf{p}$ for Y of estimated class membership probabilities computed in the way described above. Entropy is a measure of uncertainty of a random variable. Since we are looking for measures that indicate uncertainty in class membership Y, its discrete entropy is a natural choice. The discrete entropy of Y can be estimated by



(a) (b)

Fig. 4. An illustration of why entropy can be a poor estimate of classification uncertainty. The plots show estimated probability distributions for two unlabeled examples in a 10 class problem. In (a), the classifier is highly confused between classes 4 and 5. In (b), the classifier is relatively more confident that the example belongs to class 4, but is assigned higher entropy. The entropy measure is influenced by probability values of unimportant classes.

$$H(Y) = -\sum_{i=1}^{k} p_i \log(p_i). \tag{8}$$

Higher values of entropy imply more uncertainty in the distribution; this can be used as an indicator of uncertainty of an example. If an example has a distribution with high entropy, the classifier is uncertain about its class membership.

The algorithm proceeds in the following way: At each round of active learning, we compute class membership probabilities for all examples in the active pool. Examples with the highest estimated value of discrete entropy are selected to query the user. User labels are obtained and the corresponding examples are incorporated in the training set and the classifier is retrained. As will be seen in Section 4.8, active learning through entropy-based selection outperforms random selection in some cases.

4.4 Best-versus-Second Best (BvSB)

Even though EP-based active learning is often better than random selection, it has a drawback. A problem of the EP measure is that its value is heavily influenced by probability values of unimportant classes. See Fig. 4 for a simple illustration. The figure shows estimated probability values for two examples on a 10-class problem. The example on the left has a smaller entropy than the one on the right. However, from a classification perspective, the classifier is more confused about the former since it assigns close probability values to two classes. For the example in Fig. 4b, small probability values of unimportant classes contribute to the high entropy score, even though the classifier is much more confident about the classification of the example. This problem becomes even more acute when a large number of classes are present. Although entropy is a true indicator of uncertainty of a random variable, we are interested in a more specific type of uncertainty relating only to classification among the most confused classes (the example is virtually guaranteed to not belong to classes having a small probability estimate).

Instead of relying on the entropy score, we take a more greedy approach to account for the problem mentioned. We consider the difference between the probability values of the two classes having the highest estimated probability value as a measure of uncertainty. Since it is a comparison of the best guess and the second best guess, we refer to it as the BvSB approach [21]. Such a measure is a more direct way of estimating confusion about class membership from a classification standpoint. Using the BvSB measure, the example on the left in Fig. 4 will be selected to query the user. As mentioned previously, confidence estimates are reliable in the sense that classes assigned low probabilities are very rarely the true classes of the examples. However, this is only true if the initial training set size is large enough for good probability estimation. In our experiments, we start from as few as two examples for training in a 100 class problem. In such cases, initially the probability estimates are not very reliable, and random example selection gives similar results. As the number of examples in the training set grows, active learning through BvSB quickly dominates random selection by a significant margin.

4.5 Another Perspective

One way to see why active selection works is to consider the BvSB measure as a greedy approximation to entropy for estimating classification uncertainty. We describe another perspective that explains why selecting examples in this way is beneficial. The understanding crucially relies on our use of the one-versus-one approach for multiclass classification. Suppose that we wish to estimate the value of a certain example for active selection. Say its true class label is l (note that this is unknown when selecting the example). We wish to find whether the example is informative, i.e., if it will modify the classification boundary of any of the classifiers, once its label is known. Since its true label is l, it can only modify the boundary of the classifiers that separate class l from the other classes. We call these classifiers as those in contention, and denote them by $\mathcal{C}_l = \{C_{(l,i)} |\, i = 1, \ldots, k,\ i \neq l\}$, where $C_{(i,j)}$ indicates the binary classifier that separates class i from class j. Furthermore, in order to be informative at all, the selected example needs to modify the current boundary (be a good candidate for a new support vector—as indicated by its uncertainty). Therefore, one way to look at multiclass active selection for one-versus-one SVMs is the task of finding an example that is *likely to be a support vector* for one of the *classifiers in contention*, without knowing which classifiers are in contention. See Fig. 5 for an illustration.

Say that our estimated probability distribution for a certain example is denoted by $\mathbf{p}$, where p_i denotes the membership probability for class i. Also suppose that the distribution $\mathbf{p}$ has a maximum value for class h. Based on current knowledge, the most likely set of classifiers in contention is $\mathcal{C}_h$. The classification confidence for the classifiers in this set is indicated by the difference in the estimated class probability values, $p_h - p_i$. This difference is an indicator of how informative the particular example is to a certain classifier. Minimizing the difference $p_h - p_i$ or, equivalently, maximizing the confusion (uncertainty), we obtain the BvSB measure. This perspective shows that our intuition behind choosing the difference in the top two probability values of the estimated distribution has a valid underlying interpretation—it is a *measure of uncertainty for the most likely classifier in contention*. Also, the BvSB measure can then be considered to be an efficient approximation for

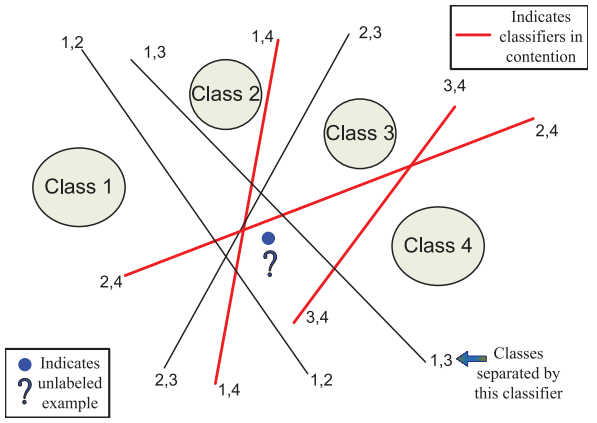


Fig. 5. Illustration of one-versus-one classification (classes that each classifier separates are noted). Assuming that the estimated distribution for the unlabeled example (shown as a blue disk) peaks at "Class 4," the set of classifiers in contention is shown as red lines. BvSB estimates the highest uncertainty in this set—uncertainty of other classifiers is irrelevant.

selecting examples that are likely to be informative, in terms of changing classification boundaries.

4.6 Binary Classification

For binary classification problems, our method reduces to selecting examples closest to the classification boundary, i.e., examples having the smallest margin. In binary problems, the BvSB measure finds the difference in class membership probability estimates between the two classes. The probabilities are estimated using (7), which relies on the function value $f(x)$ of each unlabeled example. Furthermore, the sigmoid fit is monotonic with the function value—the difference in class probability estimates is larger, for examples, away from the margin. Therefore, our active learning method can be considered to be a generalization of binary active learning schemes that select examples having the smallest margin.

4.7 Computational Cost

There are two aspects to the cost of active selection. One is the cost of training the SVM on the training set at each iteration. Second is probability estimation on the active pool and selecting examples with the highest BvSB score. Since we use one-versus-one SVM, we need to train $\mathcal{O}(k^2)$ classifiers for k classes. As the essence of active learning is to minimize training set sizes through intelligent example selection, it is also important to consider the cost of probability estimation and example selection on the relatively much larger active pool. The first cost comes from probability estimation in binary SVM classifiers. The estimation is efficient since it is performed using Newton's method with backtracking line search that guarantees quadratic rate of convergence. Given class probability values for binary SVMs, multiclass probability estimates can be obtained in $\mathcal{O}(k)$ time per example [42]. With N examples in the active pool, the entire BvSB computation scales as $\mathcal{O}(Nk^2)$.

4.8 Experiments with BvSB

In this section, we show experiments demonstrating the ability of the BvSB measure to select informative examples for query. Note that this section reports results only using this uncertainty measure and not VOI. Later, we incorporate BvSB as an approximation to reduce the computational expense of VOI computation. We demonstrate results on standard image datasets available from the UCI repository [2], the Caltech-101 dataset of object categories, and a dataset of 13 natural scene categories. All the results show significant improvement owing to active example selection. Table 2 shows a summary of datasets used and their details. For choosing the kernel, we ran supervised learning experiments with linear, polynomial, and Radial Basis Function (RBF) kernels on a randomly chosen training set, and picked the kernel that gave the best classification accuracy averaging over multiple runs.

TABLE 2
Dataset Details

Dataset	#classes	#features	# Pool	# Test	Kernel
USPS	10	256	5000	2000	Gaussian
Pendigits	10	16	5000	2000	Linear
Scene-13	13	320 [31]	5000	2000	Linear
Caltech-101	101	N/A	1515	1515	From [14]

 # pool = active pool size, # test = test set size.

4.8.1 Reduction in Training Required

In this section, we perform experiments to quantify the reduction in the number of training examples required for BvSB to obtain similar classification accuracy as random example selection. For each round of active learning, we find the number of rounds of random selection to achieve the same classification accuracy. In other words, fixing the classification accuracy achieved, we measure the difference in the training set size of both methods and report the corresponding training rounds in Table 3. The table shows that active learning achieves a reduction of about 50 percent in the number of training examples required, i.e., it can reach near optimal performance with 50 percent fewer training examples. Table 3 reports results for the USPS dataset; however, similar results were obtained for the Pendigits dataset and the Letter dataset.

An important point to note from Table 3 is that active learning does not provide a large benefit in the initial rounds. One reason for this is that all methods start with the

TABLE 3
Percentage Reduction in the Number of Training Examples Provided to the Active Learning Algorithm to Achieve Classification Accuracy Equal to or More than Random Example Selection on the USPS Dataset

BvSB selection rounds	Random selection rounds	% Reduction in # training examples
3	6	11.53
4	10	20
5	13	24.24
6	19	33.33
7	28	43.75
8	29	42.85
9	43	53.96
10	44	53.12
11	43	50.79
12	48	52.94
13	50	52.85

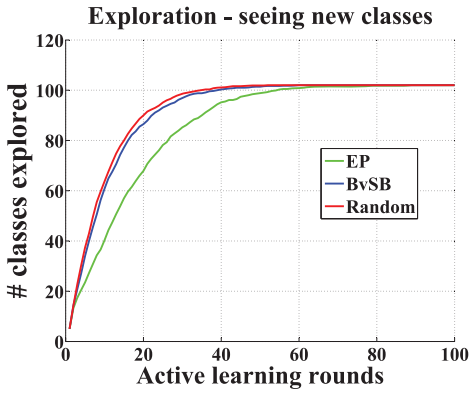


Fig. 6. Space exploration of active selection—BvSB-based selection is almost as good as random exploration, while the former achieves much higher classification accuracy than random.

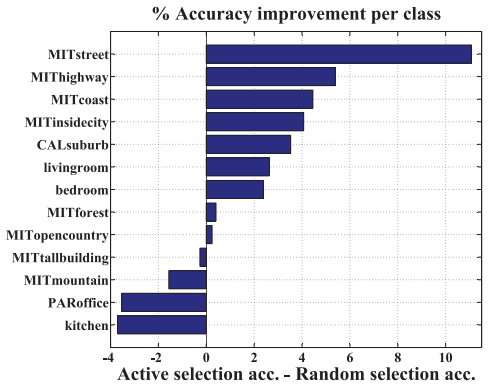


Fig. 7. Active learning on the 13 natural scene categories dataset.

same seed set initially. In the first few rounds, the number of examples actively selected is far fewer compared to the seed set size (100 examples). Actively selected examples thus form a small fraction of the total training examples, explaining the small difference in classification accuracy of both methods in the initial rounds. As the number of rounds increases, the importance of active selection becomes clear, explained by the reduction in the amount of training required to reach near-optimal performance.

4.9 Exploring the Space

In many applications, the number of categories to be classified is extremely large, and we start with only a few labeled images. In such scenarios, active learning has to balance two often conflicting objectives—exploration and exploitation. Exploration in this context means the ability to obtain labeled images from classes not seen before. Exploitation refers to classification accuracy on the classes seen so far. Exploitation can conflict with exploration since, in order to achieve high classification accuracy on the seen classes, more training images from those classes might be required while sacrificing labeled images from new classes. In the results so far, we show classification accuracy on the entire test data consisting of all classes—thus good performance requires a good balance between exploration and exploitation. Here, we explicitly demonstrate how the different example selection mechanisms explore the space for the Caltech-101 dataset that has 102 categories. Fig. 6 shows that the BvSB measure finds newer classes almost as fast as random selection, while achieving significantly higher classification accuracy than random selection. Fast exploration of BvSB implies that learning can be started with labeled images from very few classes and the selection mechanism will soon obtain images from the unseen classes. Interestingly, EP-based selection explores the space poorly.

4.9.1 Scene Recognition

Further, we performed experiments for the application of classifying natural scene categories on the 13 scene categories dataset [11]. GIST image features [31] that provide a global representation were used. Results are shown in Fig. 7. The figure shows accuracy improvement (active selection accuracy-random selection accuracy) per class after 30 BvSB-based active learning rounds. Note that although we do not explicitly minimize redundancy among images, active selection leads to significant improvements even when *as many as 20 images are selected at each active learning round*.

4.9.2 Which Examples Are Selected?

In Fig. 8, we show example images from the USPS dataset and their true labels. The top row images were confusing for the classifier (indicated by their BvSB score) and were therefore selected for active learning at a certain iteration. The bottom row shows images on which the classifier was most confident. The top row has more confusing images even for the human eye, and ones that do not represent their true label well. We noticed that the most confident images (bottom row) consisted mainly of the digits "1" and "7," which were clearly drawn. The results indicate that the active learning method selects hard examples for query.

One of the reasons active learning algorithms perform well is the imbalanced selection of examples across classes. In our case, the method chooses more examples for the classes which are hard to classify (based on how the random example selection algorithm performs on them). Fig. 9 demonstrates the imbalanced example selection across different classes on the Caltech-101 dataset. On the y-axis, we plot the number of examples correctly classified by the random example selection algorithm for each class, as an indicator of hardness of the class. Note that the test set used in this case is balanced with 15 images per class. On the x-axis, we plot the number of examples selected by the active selection algorithm for the corresponding class from the active pool. The data show a distinct negative correlation, indicating that more examples are selected from the

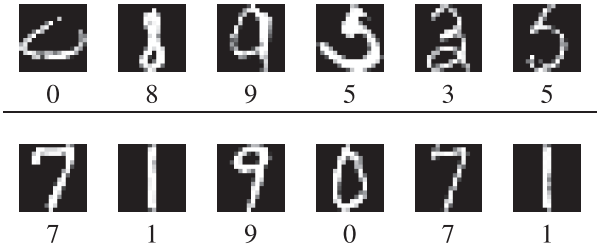


Fig. 8. The top row shows images on which the classifier is uncertain using the BvSB score. The bottom row shows images on which the classifier is confident. True labels are noted below the corresponding images. We can see that the top row has more confusing images, indicating that the active learning method chooses harder examples.

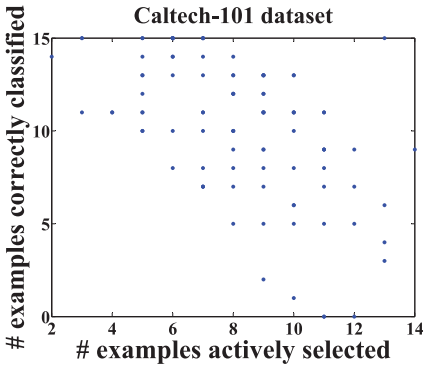


Fig. 9. Y-axis: # examples correctly classified by random example selection for a given class. X-axis: # examples of the corresponding class chosen by active selection. The negative correlation shows that active learning chooses more examples from harder classes.

harder classes, confirming our intuition. Notice the empty region on the bottom left of the figure, showing that active learning selected more images from *all* classes that were hard to classify.

4.10 Approximations to VOI

In the previous section, we showed how the proposed uncertainty sampling measure can efficiently select informative examples for active learning. Here, we discuss some approximations that substantially improve the running time of the proposed VOI algorithm using the BvSB measure. The VOI algorithm described previously in Fig. 3 is the original algorithm on which the following approximations are performed (line numbers refer to the algorithm).

4.10.1 Seed Sampling

Since VOI computation is relatively expensive, finding the scores for all examples in the active pool is costly (line 3). Instead, we use the BvSB measure to sample uncertain examples from the active pool on which VOI computation is performed. Typically, a sample of 50 examples is obtained from active pools of thousands of examples. We observed that even though BvSB and VOI do not correlate perfectly, the top 50 examples chosen by BvSB almost always contain the examples that would have been the highest ranked using VOI alone. Quantitatively, the results differ only 2 percent of the time, and the difference in classification accuracy is negligible. On the other hand, the computational speedups achieved are substantial.

4.10.2 Expected Value Computation

In the VOI algorithm, estimating expected risk is expensive. For each unlabeled image, we need to train classifiers assuming that the image can belong to any of the possible categories (line 4). This can be slow when many classes are present. To overcome this, we make the following observation: Given the estimated probability distribution of an unlabeled image, it is unlikely to belong to the classes that are assigned low probability values, i.e., the image most likely belongs to the classes that have the highest estimated probabilities. As such, instead of looping over all possible classes, we can only loop over the most likely ones. In particular, we loop over only the top two most likely classes as they contain most of the discriminative information as utilized in the BvSB measure. Such an approximation relies to some extent on the correctness of the estimated model, which implies an *optimistic* assumption often made for computational tractability [15]. Further, we can use the same "top-2" approximation for computing the expected risk (line 9) on unlabeled images as an approximation to (1).

4.10.3 Clustering for Estimating Risk

In the above algorithm, the risk needs to be estimated on the entire active pool. Instead, we first cluster the unlabeled images in the active pool using the kernel k-means algorithm [45]. Then, we form a new unlabeled image set by choosing one representative (closest to the centroid) image from each cluster, and estimate risk on this reduced set. The clustering needs to be performed only once initially, and not in every query iteration. In our implementation, we fix the number of clusters as $1/100$ fraction of the active pool size. Experiments showed that this approximation rarely (less than 5 percent of the time) changes the images selected actively, and makes a negligible difference in the estimated risk value and the future classification accuracy.

With the above approximations, the complexity of each query iteration is $\mathcal{O}(Nk^2)$, a large improvement over the original version. In Section 4, we propose a sublinear time approximation for scaling to very large datasets.

5 Experimental Results

In this section, we evaluate the proposed VOI algorithm on various datasets described in Table 2. Scene-13 is a dataset of 13 natural scene categories [11], for which we employ GIST features [31]. Precomputed pyramid match kernel matrices [14] were used as features for the Caltech-101 dataset.

For implementation we used Matlab along with the LIBSVM toolbox [5] (written in C, interfaced with Matlab for SVM and probability estimation). With an active pool size of 5,000 images for a 10-class problem (USPS), each query iteration on average takes about 0.9 seconds on a 2.67 Ghz Xeon machine. For the Caltech dataset with an active pool of size 1,515 images with 101 classes, a query iteration takes about 1.3 seconds.

5.1 User Interaction Time

We have previously demonstrated the benefits of the BF model as compared to MCF from the ease of interaction standpoint. Here, we compare the total user annotation time required with various methods to achieve similar classification rates. The comparison shows the following methods: our proposed VOI method with binary feedback (VOI+BF), VOI with MCF, active learning using only the BvSB measure (US+MCF), where US stands for uncertainty sampling, and random selection with both BF and MCF. Fig. 10 shows the substantial reduction in user training time with the proposed method. For all the datasets, the proposed VOI-based algorithm beats all others (including active selection with MCF), indicating that the advantages come from both **our active selection algorithm, as well as the binary feedback model**. Further, note that the relative improvement is larger for the Caltech dataset as it has a larger number of categories. As such, we can train classifiers in a fraction of the time typically

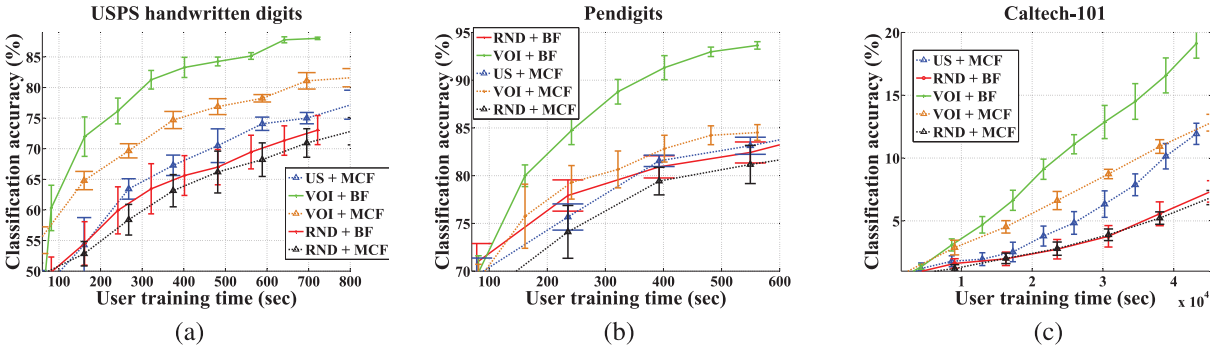


Fig. 10. Active learning in the BF model requires far less user training time compared to active selection in the MCF model. US: uncertainty sampling, RND: random. (a) USPS, (b) Pendigits, (c) Caltech-101 datasets.

required, demonstrating the strength of our approach for multiclass problems.

5.2 Importance of Considering Annotation Cost

As mentioned before, we use uncertainty sampling (US)-based active selection to form a smaller set from which the most informative images are selected using VOI computation. Here, we demonstrate that the good results are not due to uncertainty sampling alone. Fig. 11 compares the *number of binary comparisons the user has to provide* in our algorithm along with the BvSB uncertainty sampling method (also in the BF model) in the initial stages of active learning. The figure shows two plots with 50 and 70 class problems, obtained from the Caltech-101 dataset. Our method significantly outperforms US in both cases, and the relative improvement increases with problem size. As the number of classes increases, considering user annotation cost for each query image becomes increasingly important. The VOI framework captures annotation cost unlike US, explaining the better performance for the 70 class problem.

5.3 Active Selection (VOI) versus Random Selection

Fig. 12 shows the confusion matrices for active selection with VOI as well as random selection on the Caltech 101 class problem. Active selection results in much less confusion, also indicated by the trace of the two matrices. This demonstrates that the algorithm offers large advantages for many category problems. Fig. 14 shows per-class classification accuracy of both VOI and random selection methods on the Scene-13 dataset. VOI achieves higher accuracy for 9 of the 13 classes, and comprehensively beats random selection in the overall accuracy.

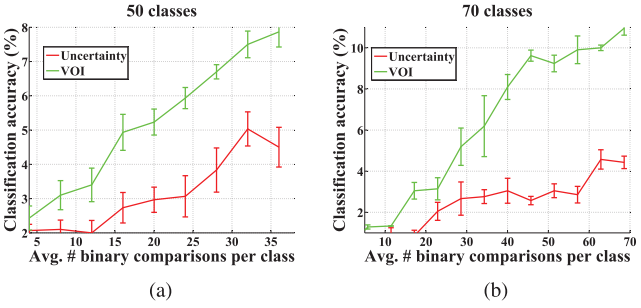


Fig. 11. VOI-based active selection and uncertainty sampling (both with BF) during the initial phases of active learning.

5.4 Noise Sensitivity

In many real-world learning tasks, the labels are noisy, either due to errors in the gathering apparatus or even because of human annotation mistakes. It is therefore important for the learning algorithm to be robust to a reasonable amount of labeling noise. In this section, we perform experiments to quantify the noise sensitivity of the methods. We artificially impart stochastic labeling noise to the training images. For example, 5 percent noise implies that training images are randomly given an incorrect label with a probability of 0.05. The algorithms are then run on the noisy as well as clean data—results for the USPS dataset are shown in Fig. 13.

The figure shows both active and random selection on clean as well as noisy data (10 and 20 percent noise). Expectedly, there is a reduction in classification accuracy for both algorithms when noise is introduced. Interestingly, however, even with as much as 10 percent label noise, the active learning method still outperforms random selection on clean data, whereas with about 20 percent noise, active

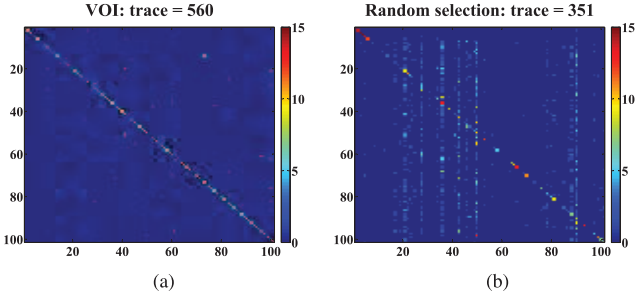


Fig. 12. Confusion matrices with (a) active (VOI), and (b) random selection (max. trace = 1,515). VOI leads to much lower confusion.

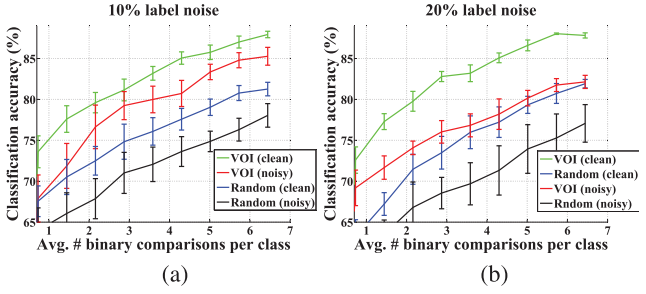


Fig. 13. Sensitivity to label noise, (a) 10 percent, (b) 20 percent. VOI with noisy data outperforms the random selection with clean data.

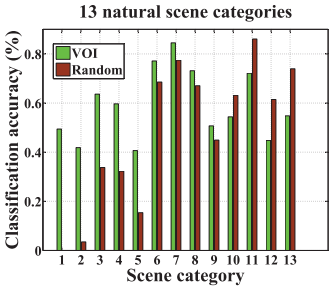


Fig. 14. Per-class accuracy of VOI versus random on the scene-13 dataset.

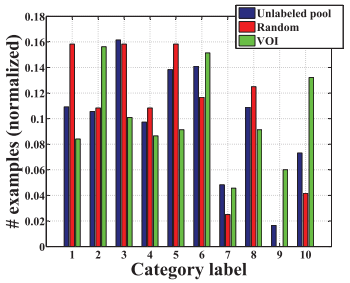


Fig. 15. Population imbalance: VOI selects many images, even for classes with small populations (see text for details).

learning still matches random selection on clean data. This result shows that active selection can tolerate a significant amount of noise while giving a high classification rate.

One reason why active selection can be robust to noise arises from the fact that the algorithm selects "hard" examples for query. In most cases, these examples lie close to the separating boundaries of the corresponding classifiers. Intuitively, we expect noise in these examples to have a smaller effect since they change the classification boundary marginally. In contrast, a misclassified example deep inside the region associated with a certain class can be much more harmful. In essence, through its example selection mechanism, active learning encounters noise that has a relatively smaller impact on the classification boundary, and thus the future classification rate.

5.5 Population Imbalance

Real-world data often exhibits class population imbalance, with vastly varying number of examples belonging different classes [10]. For example, in the Caltech-101 dataset, the category "airplanes" has over 800 images, while the category "wrench" has only 39 images.

We demonstrate here that active selection can effectively counter population imbalances in order to generalize better. The experiment is conducted as follows: The active pool (from which unlabeled images are selected for query) consisting of a vastly varying number of examples of each class is generated for the Pendigits dataset. However, the test set is kept unmodified. In this scenario, random example selection suffers since it obtains fewer examples from the less populated classes. Active selection, on the other hand, counters the imbalance by selecting a relatively higher number of examples even from the less populated classes. Fig. 15 demonstrates the results. The three bars show (normalized) number of examples per class in the unlabeled pool, and in the training sets with active and random selection. Random selection does poorly—for instance, it does not obtain even a single training image from class "9" due to its low population in the unlabeled pool. Active selection overcomes population imbalance and selects many images from class "9." This is further reinforced by computing the variance in the normalized population. The standard deviation in the (normalized) number of examples selected per class with active and random selection are 0.036 and 0.058, respectively. The significantly smaller deviation shows that active selection overcomes population imbalance to a large extent.

5.6 Fast Initiation of New Classes

In Section 3.4, we described our method of quickly initiating new classes and then merging the erroneous ones using agglomerative clustering and user feedback. Table 4 summarizes the advantages of the approach (i.e., with clustering) compared to simple category initiation when a new image does not match any training image (naive). We start with a small seed set of 20 images and run the experiment until both methods encounter all 101 categories in the data. Note the large reduction in user training time with clustering due to the fewer number of binary comparisons requested. This aspect is increasingly important as the number of classes increases.

6 Speeding Up Active Learning

There has been some recent work on scaling up active learning to work with large datasets. In [44], a graph-regularization approach is proposed to maximize the expected information gain for scalable active learning. Segal et al. [35] propose an approximate uncertainty sampling approach in which only a subset of samples are evaluated at each iteration for active learning. Their approach provides speedups for the application of labeling e-mail corpora. A hierarchical sampling approach along with feature space indexing was proposed for scaling active learning to large datasets by Panda et al. [32].

In this section, we show initial results with a different approach to speeding up active learning via locality sensitive hashing (LSH) [19]. As opposed to previous work, our method does not modify the active selection criteria and can work with any classifiers. Instead of performing an exhaustive search with a linear scan (LS) of the entire unlabeled pool, the main idea is to first find representative samples that are informative (for seeding the search) according to our active selection measure. Using locality sensitive hashing on these samples, informative samples from the unlabeled pool are obtained (in time scaling sublinearly with the pool size). This approach provides *up to two orders of magnitude speed-up* on the linear scan active learning version, while making little difference in classification accuracy. We can thus scale the algorithm to datasets

TABLE 4
User Training Time Required to Encounter All 101 Classes

Dataset	W/ clustering	Naive
Caltech-101	2560 sec	3200 sec

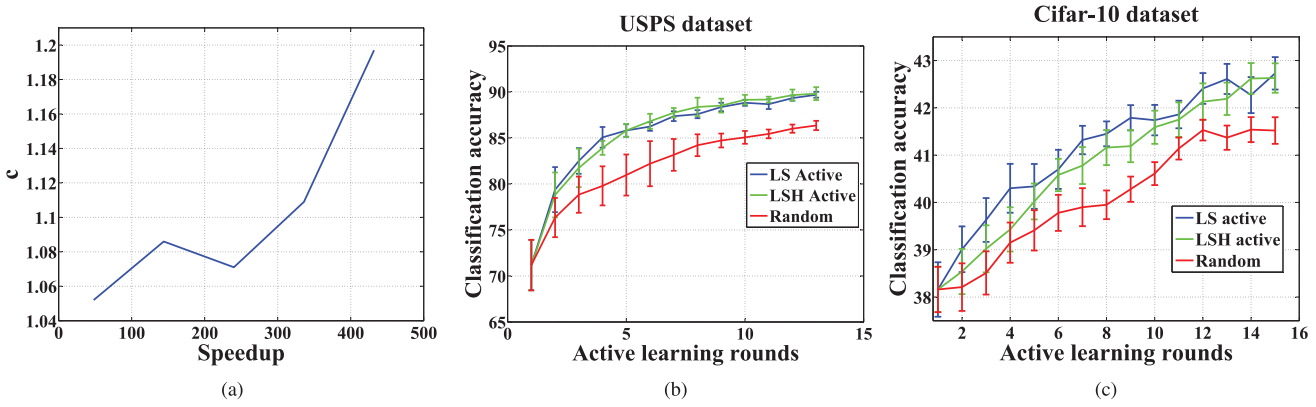


Fig. 16. (a) Speedup achieved with LSH over LS for the approximate near neighbor problem on the Cifar-10 dataset. $c = 1 + \epsilon$ denotes the approximation factor. (b), (c) Active learning with the LSH approximation gives little difference in accuracy compared to the LS. (b) USPS dataset, (c) Cifar-10 dataset. On average, **the speedup for USPS was 17-fold, while that for Cifar-10 was 91-fold.**

with hundreds of thousands of samples. With a pool size of 50,000 images represented in 384-dimensional space, the LSH-based approximation provides a *91-fold speedup on average* with negligible reduction in classification accuracy. In the following, we provide a brief introduction to LSH using p-stable distributions [6], followed by its application in our active learning algorithm.

6.1 LSH with p-Stable Distributions

Definition 1 [19]. *An LSH family $\mathcal{H} = \{h : S \rightarrow U\}$ is called (r_1, r_2, p_1, p_2)-sensitive for distance d if for any $u, v \in S$,*

- *if $u \in B(v, r_1)$, then $Pr_{\mathcal{H}}[h(u) = h(v)] \geq p_1$,*
- *if $u \notin B(v, r_2)$, then $Pr_{\mathcal{H}}[h(u) = h(v)] \leq p_2$,*

where $B(q, r)$ indicates a ball of radius r centered at q. If $p_1 > p_2$ and $r_1 < r_2$, the family $\mathcal{H}$ can be used for the (R, c)-NN problem [19] wherein one has to retrieve points p such that $d(p, q) \leq cR$ if there exists a point in P within distance R from q. The basic idea is that the hash functions evaluate to the same values with high probability for points that are close to each other, whereas for distant points the probability of matching (collision) is low. The probability gap can be increased by concatenation of multiple hash functions chosen randomly from the family $\mathcal{H}$.

6.2 Sublinear Time Active Learning

Here, we propose a simple way to speed up active learning using LSH. During preprocessing, we first hash all the points in the database[3] to the respective buckets using the chosen hash functions. At each iteration, we pick the samples from our training data that give the highest VOI assuming they are unlabeled. These samples are treated as *informative seed samples* that will be used as queries to retrieve the nearest neighbors from the active pool, in the hope that they will also be informative. Since the training set is usually orders of magnitude smaller than the unlabeled pool, a linear scan to choose best samples from it does not slow down the algorithm. Also, other seeding strategies that do not require a scan could easily be employed instead.

3. It is shown in [6] that the hash functions $h_{\mathbf{a},b}(\mathbf{x}) = \lfloor \frac{\mathbf{a} \cdot \mathbf{x} + b}{r} \rfloor$, where each element of $\mathbf{a}$ is sampled from $\mathcal{N}(0, 1)$ and b chosen uniformly from $[0, r]$, represents an (R, cR, p_1, p_2)-sensitive LSH family for the euclidean distance measure.

Assuming that the VOI function is spatially smooth, the rationale behind choosing the nearest neighbors of the points with high VOI is to find other *unlabeled points with high VOI*. Intuitively, many functions that capture informativeness of samples, such as distance to hyperplane, etc., can be reasonably assumed to be smooth so that such a search will lead to useful samples for active learning. Furthermore, note that the proposed strategy does not depend on the choice of the classifier or the active selection measure used. It can be employed for other classifiers as well as other selection measures seamlessly. The hashing method as proposed requires the explicit feature vectors of the data samples, and as such cannot be used directly for kernel matrices. Extending to kernels using kernelized LSH [27] is an interesting direction for future work.

6.3 Experiments with Hashing

Experiments are performed on two datasets: the USPS dataset used previously and the Cifar-10 dataset [26], which is a collection of 50,000 training images and 10,000 test images obtained from the 80 million tiny images dataset [40]. For Cifar-10, 384-d GIST descriptors [31] are used as per [26].

Our algorithm relies on LSH retrieving points that are close to the query points with high probability. Here, we first perform an experiment with the Cifar-10 dataset to analyze how efficiently nearest neighbors are retrieved by LSH. The setup is as follows: For each iteration, a random point was selected as the query. The LSH and LS were run to find the near neighbors of the query, while noting the time required for both along with the distance to the nearest neighbor found (LS finds the true nearest neighbor). The distance to the nearest neighbor found by LSH is normalized by the distance to the true neighbor to find the approximation factor $c = 1 + \epsilon$. We ran 1,000 such iterations and the resulting speedup values were put into five bins.

Fig. 16a shows a plot of the approximation factor achieved versus speedup. As expected, we see that a higher speedup gives worse approximation. The speedups, however, are large across the entire spectrum of approximation values, achieving a 400-fold speedup for a 0.2-approximation ($c = 1.2$). Note that the approximation guarantees for LSH are conservative, and we observe significantly better performance in practice. Furthermore, since the LSH

algorithm scales sublinearly with data size, we expect the speedups to be even larger for bigger datasets.

It is important to note that even a crude approximation to nearest neighbor does not necessarily hurt active learning. Active selection measures are typically based on computations of potential informativeness of the data sample which are often approximate, and are heavily dependent on the current model. As such, even points that are not the nearest neighbors to informative queries might have very close (and sometimes even better) informativeness scores than the true nearest neighbors. Our experiment below demonstrates that this is indeed the case: An approximate nearest neighbor often makes no difference in the informativeness values of the chosen samples as well as in the final classification accuracy achieved.

Figs. 16b and 16c show classification accuracy comparisons between LS and LSH active learning algorithms. In both plots, the difference in accuracy due to the approximation is very small, whereas the LSH-based active learning algorithms run about 1 and 2 orders of magnitude faster, respectively, on USPS ($\sim 5{,}000$ samples) and Cifar-10 ($\sim 50{,}000$ samples). As mentioned before, the speedup is expected to increase with the dataset size since the linear scan takes $\mathcal{O}(N)$ time, whereas LSH-based active learning runs in expected time $\mathcal{O}(N^{\rho})$ with $\rho < 1$. This demonstrates the powerful scaling ability of the locality sensitive hashing approach to active learning.

7 Conclusion and Future Work

In this paper, we proposed a multiclass active learning framework using only binary feedback. A value of information algorithm was developed for active learning in this framework, along with a simple and efficient active selection measure. The feedback modality allows very efficient annotation in multiclass problems and thereby substantially reduces training time and effort. Further, we presented results using locality sensitive hashing to speed up active learning so as to achieve sublinear time scaling (w.r.t. dataset size) for choosing a query. The proposed modification achieved two orders of magnitude speedup with little difference in classification accuracy. Future work will focus on batch-mode sampling and further improving scaling to allow thousands of data categories along with millions of samples.

Acknowledgments

This work was supported in part by the US National Science Foundation (NSF) through grants #IIP-0443945, #IIP-0726109, #CNS-0708344, #CNS-0821474, #IIP-0934327, #CNS-1039741, #IIS-1017344, #IIP-1032018, and #SMA-1028076, the Minnesota Department of Transportation, and the ITS Institute at the University of Minnesota. The authors thank Professor Kristen Grauman for providing kernel matrices for Caltech-101 data, and the anonymous reviewers for their helpful suggestions.

References

[1] "Amazon Mechanical Turk," http://www.mturk.com, 2012.

[2] A. Asuncion and D.J. Newman, "UCI Machine Learning Repository," Univ. of California, Irvine, School of Information and Computer Sciences, http://archive.ics.uci.edu/ml/datasets.html, 2007.

[3] S. Basu, A. Banerjee, and R. Mooney, "Semi-Supervised Clustering by Seeding," *Proc. 19th Int'l Conf. Machine Learning,* 2002.

[4] C. Campbell, N. Cristianini, and A.J. Smola, "Query Learning with Large Margin Classifiers," *Proc. Seventh Int'l Conf. Machine Learning,* 2000.

[5] C.-C. Chang and C.-J. Lin, *LIBSVM: A Library for Support Vector Machines,* software available at http://www.csie.ntu.edu.tw/cjlin/libsvm, 2001.

[6] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni, "Locality-Sensitive Hashing Scheme Based on p-Stable Distributions," *Proc. 20th Ann. Symp. Computational Geometry,* 2004.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A Large-Scale Hierarchical Image Database," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[8] C.H.Q. Ding, X. He, H. Zha, M. Gu, and H.D. Simon, "A Min-Max Cut Algorithm for Graph Partitioning and Data Clustering," *Proc. IEEE Int'l Conf. Data Mining,* 2001.

[9] K.-B. Duan and S.S. Keerthi, "Which Is the Best Multi-Class SVM Method? An Empirical Study," *Proc. Sixth Int'l Workshop Multiple Classifier Systems,* 2005.

[10] S. Ertekin, J. Huang, L. Bottou, and L. Giles, "Learning on the Border: Active Learning in Imbalanced Data Classification," *Proc. 16th ACM Conf. Information and Knowledge Management,* 2007.

[11] L. Fei-Fei and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2005.

[12] L. Fei-Fei, P. Perona, and R. Fergus, "One-Shot Learning of Object Categories," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 4, pp. 594-611, Apr. 2006.

[13] Y. Freund, H.S. Seung, E. Shamir, and N. Tishby, "Selective Sampling Using the Query by Committee Algorithm," *Machine Learning,* vol. 28, pp. 133-168, 1997.

[14] K. Grauman and T. Darrell, "The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features," *Proc. 10th IEEE Int'l Conf. Computer Vision,* 2005.

[15] Y. Guo and R. Greiner, "Optimistic Active Learning Using Mutual Information," *Proc. 20th Int'l Joint Conf. Artifical Intelligence,* 2007.

[16] S.C. Hoi, R. Jin, J. Zhu, and M.R. Lyu, "Semi-Supervised SVM Batch Mode Active Learning for Image Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[17] A. Holub, P. Perona, and M. Burl, "Entropy-Based Active Learning for Object Recognition," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition Workshops,* 2008.

[18] C.-W. Hsu and C.-J. Lin, "A Comparison of Methods for Multi-Class Support Vector Machines," *IEEE Trans. Neural Networks,* vol. 13, no. 2, pp. 415-425, Mar. 2002.

[19] P. Indyk and R. Motwani, "Approximate Nearest Neighbor: Towards Removing the Curse of Dimensionality." *Proc. Symp. Theory of Computing,* 1998.

[20] P. Jain and A. Kapoor, "Active Learning for Large Multi-Class Problems," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[21] A.J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-Class Active Learning for Image Classification," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[22] A.J. Joshi, F. Porikli, and N. Papanikolopoulos, "Breaking the Interactive Bottleneck in Multi-Class Classification with Active Selection and Binary Feedback," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[23] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Active Learning with Gaussian Processes for Object Categorization," *Proc. 11th IEEE Int'l Conf. Computer Vision,* 2007.

[24] A. Kapoor, E. Horvitz, and S. Basu, "Selective Supervision: Guiding Supervised Learning with Decision-Theoretic Active Learning," *Proc. IEEE Eighth Int'l Conf. Data Mining,* 2007.

[25] A. Krause and C. Guestrin, "Near-Optimal Nonmyopic Value of Information in Graphical Models," *Proc. 21st Ann. Conf. Uncertainty in Artificial Intelligence,* 2005.

[26] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," technical report, Univ. of Toronto, 2009.

[27] B. Kulis and K. Grauman, "Kernelized Locality-Sensitive Hashing for Scalable Image Search," *Proc. 12th IEEE Int'l Conf. Computer Vision,* 2009.

[28] M. Li and I. Sethi, "Confidence-Based Active Learning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 8, pp. 1251-1261, Aug. 2006.
[29] H.-T. Lin, C.-J. Lin, and R.C. Weng, "A Note on Platt's Probabilistic Outputs for Support Vector Machines," *Machine Learning,* vol. 68, pp. 267-276, 2007.
[30] T. Mitchell, *Machine Learning.* McGraw-Hill, 1997.
[31] A. Oliva and A. Torralba, "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope." *Int'l J. Computer Vision,* vol. 42, no. 3, pp. 145-175, 2001.
[32] N. Panda, K. Goh, and E. Chang, "Active Learning in Very Large Image Databases," *J. Multimedia Tools and Applications,* special issue on computer vision meets databases, vol. 31, no. 3, pp. 249-267, 2006.
[33] J. Platt, "Probabilistic Outputs for Support Vector Machines and Comparison to Regularized Likelihood Methods," *Advances in Large Margin Classifiers,* MIT Press, 2000.
[34] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, and H.-J. Zhang, "Two-Dimensional Active Learning for Image Classification," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.
[35] R. Segal, T. Markowitz, and W. Arnold, "Fast Uncertainty Sampling for Labeling Large E-Mail Corpora," *Proc. Conf. Email and Anti-Spam,* 2006.
[36] B. Settles, "Active Learning Literature Survey," Computer Sciences Technical Report 1648, Univ. of Wisconsin–Madison, 2009.
[37] A. Sorokin and D. Forsyth, "Utility Data Annotation with Amazon Mechanical Turk," *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops,* 2008.
[38] S. Tong and E. Chang, "Support Vector Machine Active Learning for Image Retrieval," *Proc. Ninth ACM Int'l Conf. Multimedia,* 2001.
[39] S. Tong and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," *J. Machine Learning Research,* vol. 2, pp. 45-66, 2001.
[40] A. Torralba, R. Fergus, and W.T. Freeman, "80 Million Tiny Images: A Large Database for Non-Parametric Object and Scene Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 11, pp. 1958-1970, Nov. 2008.
[41] S. Vijayanarasimhan and K. Grauman, "What's It Going to Cost You?: Predicting Effort vs. Informativeness for Multi-Label Image Annotations," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.
[42] T.-F. Wu, C.-J. Lin, and R.C. Weng, "Probability Estimates for Multi-Class Classification by Pairwise Coupling," *J. Machine Learning Research,* vol. 5, pp. 975-1005, 2004.
[43] R. Yan, J. Yang, and A. Hauptmann, "Automatically Labeling Video Data Using Multi-Class Active Learning," *Proc. Ninth IEEE Int'l Conf. Computer Vision,* pp. 516-523, 2003.
[44] W. Zhao, J. Long, E. Zhu, and Y. Liu, "A Scalable Algorithm for Graph-Based Active Learning," *Proc. Second Ann. Int'l Workshop Frontiers in Algorithmics,* 2008.
[45] J. Shaw-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis.* Cambridge Univ. Press, 2004.

Ajay J. Joshi received the BE degree in instrumentation and control engineering from the Government College of Engineering Pune in 2005, the MSEE degree in electrical engineering from the University of Minnesota in 2008, and the PhD degree in computer science from the University of Minnesota in 2011. He currently works as a software engineer at Google. His interests include machine learning, computer vision, pattern recognition, and data mining. He was a Doctoral Spotlight presenter at CVPR 2009, a finalist for the Microsoft Research PhD Fellowship in 2009, and a Graduate School Doctoral Dissertation Fellow at the University of Minnesota in 2010-2011. He is a member of the IEEE.

Fatih Porikli received the PhD degree from New York University and, before joining Mitsubishi Electric Research Labs (MERL) in 2000, he developed satellite imaging solutions at Hughes Research Labs and 3D capture and display systems at AT&T Research Labs. He is currently a distinguished scientist at MERL, Cambridge, Massachusetts. His work covers areas including computer vision, machine learning, sparse reconstruction, video surveillance, multimedia denoising, biomedical vision, radar signal processing, online learning, etc. He received the R&D100 2006 Award in the Scientist of the Year category (select group of winners) in addition to numerous best paper and professional awards. He serves as an associate editor for many IEEE, Springer, Elsevier, and SIAM journals. He was the general chair of IEEE AVSS '10 and is an organizer of several other IEEE conferences. He is a senior member of the IEEE.

Nikolaos P. Papanikolopoulos received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Greece, in 1987, the MSEE degree in electrical engineering from Carnegie Mellon University (CMU), Pittsburgh, Pennsylvania, in 1988, and the PhD degree in electrical and computer engineering from Carnegie Mellon University in 1992. Currently, he is a Distinguished McKnight University Professor in the Department of Computer Science at the University of Minnesota and the director of the Center for Distributed Robotics and SECTTRA. His research interests include robotics, computer vision, sensors for transportation applications, and control. He has authored or coauthored more than 280 journal and conference papers in the above areas (67 refereed journal papers). He was a finalist for the Anton Philips Award for Best Student Paper at the 1991 IEEE International Conference on Robotics and Automation and the recipient of the Best Video Award at the 2000 IEEE International Conference on Robotics and Automation. Furthermore, he was a recipient of the Kritski fellowship in 1986 and 1987. He was a McKnight Land-Grant Professor at the University of Minnesota for the period 1995-1997 and has received the US National Science Foundation (NSF) Research Initiation and Early Career Development Awards. He was also awarded the Faculty Creativity Award by the University of Minnesota. One of his papers (coauthored by O. Masoud) was awarded the IEEE VTS 2001 Best Land Transportation Paper Award. Finally, he has received grants from the US Defense Advanced Research Projects Agency (DARPA), DHS, US Army, US Air Force, Sandia National Laboratories, NSF, Lockheed Martin, Microsoft, INEEL, USDOT, MN/DOT, Honeywell, and 3M (more than $20M). He is a fellow of the IEEE.