# Breaking the interactive bottleneck in multi-class classification with active selection and binary feedback

Ajay J. Joshi
University of Minnesota
Twin Cities
ajay@cs.umn.edu

Fatih Porikli
Mitsubishi Electric Research
Laboratories
fatih@merl.com

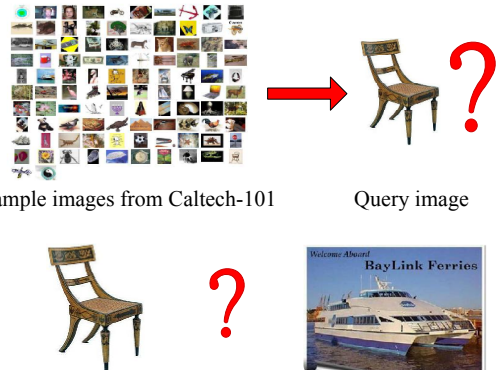
Nikolaos Papanikolopoulos
University of Minnesota
Twin Cities
npapas@cs.umn.edu

## Abstract

*Multi-class classification schemes typically require human input in the form of precise category names or numbers for each example to be annotated – providing this can be impractical for the user when a large (and possibly unknown) number of categories are present. In this paper, we propose a multi-class active learning model that requires only binary (yes/no type) feedback from the user. For instance, given two images the user only has to say whether they belong to the same class or not. We first show the interactive benefits of such a scheme with user experiments. We then propose a Value of Information (VOI)-based active selection algorithm in the binary feedback model. The algorithm iteratively selects image pairs for annotation so as to maximize accuracy, while also minimizing user annotation effort. To our knowledge, this is the first multi-class active learning approach that requires only yes/no inputs. Experiments show that the proposed method can substantially minimize user supervision compared to the traditional training model, on problems with as many as 100 classes. We also demonstrate that the system is robust to real-world issues such as class population imbalance and labeling noise.*

## 1. Introduction

In most image classification problems, we typically have a large number of unlabeled images. Intelligently exploiting the large amounts of data is a challenging problem. Active learning aims to select informative images from large data to train classifiers, and has received substantial interest for binary [21] and recently even multi-class classification settings [25, 11, 13, 12, 14, 23]. Even though multi-class 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: (i) 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; (ii) the time and effort required increase with an increase in the number of categories; (iii) the interaction is prone to mistakes in annotation, and (iv) it is not easily amenable to distributed annotation as all users need to be consistent in labeling.




Sample images from Caltech-101 Query image

Figure 1. Top row: sample interaction in traditional multi-class 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.

Image datasets are ever increasing in their size and the image variety - it is not uncommon to have thousands of image classes [4, 22]. 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. Towards this objective, we propose here a general framework for multi-class active learning that requires only yes/no feedback from the user. A simple illustration of the different interaction models is depicted in Figure 1. During each instance of interaction, the user is presented with two images and has to say whether those images belong to the same category or not. Giving such input is extremely easy, and since only two images need to be compared every time, it is also less prone to human mistakes. It easily allows distributed annotation as well.

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

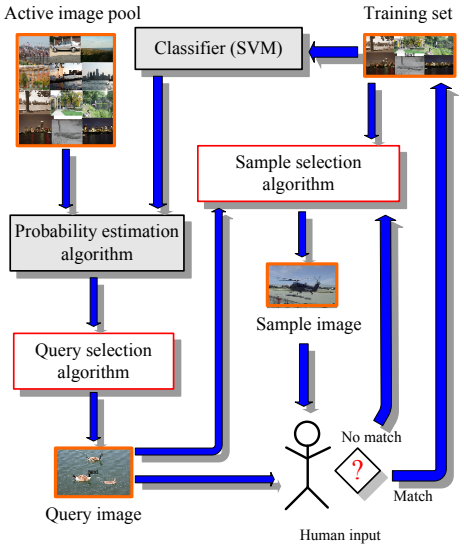



Figure 2. Block schematic of the active learning setting. Our focus in this paper is on the query and sample selection algorithms – depicted in white boxes with red borders (see text for details).

### 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 [8]. Each user was asked to interact with two modalities: i) giving category labels (out of a given set of labels) to randomly queried images, as is typically used for training, and ii) 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.

| 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 |

Table 1. Comparing the two interaction modalities.

First, it can be seen that binary feedback (BF) requires far lesser user time than giving multi-class 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 [10]), or when notions of categories change with time.

In summary, binary feedback provides an extremely appealing interaction model for large problems with many classes. In Section 4, we also quantitatively demonstrate the benefits of the binary model through experiments.

### 1.2. Learning setup

Figure 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 multi-class case, one-vs-one SVM (classifiers trained for each pair of classes) are used.

In the traditional multi-class 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 Figure 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 [19, 17] to estimate binary probabilities based on the SVM margins, combined with pairwise coupling [24] with one-vs-one SVM for multi-class probability estimation on the unlabeled images.

In Figure 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[1]. 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 [2]. 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, we demonstrate empirically that the setup we employ is robust to labeling noise.

## 2. The active learning method

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

### 2.1. Query selection

The goal here is to query informative images, i.e., images that are likely lead to an improvement in future classification accuracy. We use the Value of Information framework [16, 15, 23] 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.

[1]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.

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

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

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

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

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

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

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 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 value of information (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:

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

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 $M_{ij} = 1, \text{ for } i \neq j$.

## 2.2. 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 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 $p_1 \geq p_2 \geq \ldots \geq p_k$. 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.

### 2.2.1 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

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

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.

## 2.3. 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.

### 2.3.1 Initiating new classes

Many active learning methods make the restrictive assumption that the initial training set contains examples from all categories [10]. 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.

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

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

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

Figure 3. Multi-class active learning with binary feedback.

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 [5], 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. Computational considerations

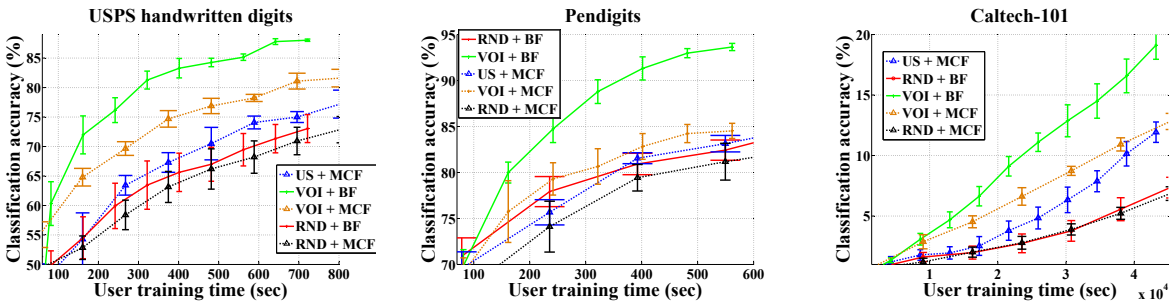
The computational complexity of each query iteration in our algorithm (Figure 3) is $\mathcal{O}(N^2k^3)$, with an active pool of size $N$ and $k$ classes. Although it works well for small problems, the cost can be impractical at larger scales. In this section, we use some approximations to significantly reduce the computational expense, and make the implementation efficient for large problems with many classes.

### 3.1. Expected value computation

In the above 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 2 most likely classes, as they contain most of the discriminative information, as noted in [13], while the smaller probability values contain little information. Such an approximation relies to some extent on the correctness of the estimated model, which implies an *optimistic* assumption often made for computational tractability [10]. Further, we can use the same "top-2" approximation, for computing the expected risk (line 9) on unlabeled images, as an approximation to Eqn. (1).

### 3.2. 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 [20]. 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% of the




(a) (b) (c)

Figure 4. Active learning in the BF model requires far lesser user training time compared to active selection in the MCF model. US: uncertainty sampling, RND: random. (a) USPS, (b) Pendigits, (c) Caltech-101 datasets.

time) changes the images selected actively, and makes a negligible difference in the estimated risk value, and the future classification accuracy.

Another approximation used is sampling of examples from the active pool to obtain a smaller set on which VOI computation is performed. Efficient active selection heuristics such as uncertainty sampling can be exploited to form the small set. Using the uncertainty sampling algorithm from [13], we first sample a small set of about 50 images from the active pool, and then select the images from this smaller set using VOI.

With the above approximations, the complexity of each query iteration is $\mathcal{O}(Nk^2)$, a large improvement over the original version. This is much better than the often observed cubic scaling for active selection [11, 14]. Supplementary material has more details about algorithm complexity.

## 4. Experiments

In this section, we evaluate the proposed algorithm on various datasets and compare it with other learning modalities. Table 2 summarizes the datasets used for experiments. USPS and Pendigits datasets were obtained from the UCI repository [1]. Scene-13 is a dataset of 13 natural scene categories [7], for which we employ GIST features [18]. Precomputed pyramid match kernel matrices [9] were used as features for the Caltech-101 dataset.

For implementation we used Matlab along with the LIBSVM toolbox [3] (written in C, interfaced with Matlab for SVM and probability estimation). With an active pool size of 5000 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 1515 images with 101 classes, a query iteration takes about 1.3 seconds. This time is lesser than the average amount of time taken by the user to give binary feedback (see Table 1). Thus, computation time is not a bottleneck and the system is interactively appealing.

### 4.1. User interaction time

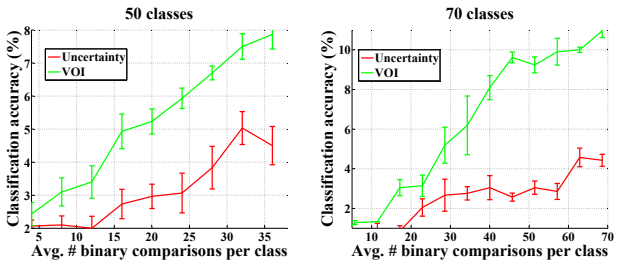
We have previously demonstrated the benefits of the BF model as compared to MCF from the ease of interaction standpoint. Here we compare the total user annotation time required with various methods to achieve similar classification rates. The comparison shows the following methods: our proposed VOI method with binary feedback (VOI+BF), VOI with MCF, active learning using the uncertainty sampling method in [13] (US+MCF), and random selection with both BF and MCF. Figure 4 (figures best viewed in color) 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 required, demonstrating the strength of our approach for multi-class problems.

| Dataset | #classes | #features | # Pool | # Test | Kernel |
|---|---|---|---|---|---|
| USPS | 10 | 256 | 5000 | 2000 | Gaussian |
| Pendigits | 10 | 16 | 5000 | 2000 | Linear |
| Scene-13 | 13 | 320 [18] | 5000 | 2000 | Linear |
| Caltech-101 | 101 | N/A | 1515 | 1515 | From [9] |

Table 2. Dataset details. # pool = active pool size, # test = test set size.

### 4.2. 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. Figure 5 compares the *number of binary comparisons the user has to provide* in our algorithm along with the 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.




(a) (b)

Figure 5. VOI-based active selection and uncertainty sampling (both with BF) during the initial phases of active learning.

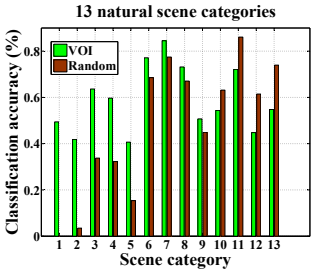



Figure 7. Per-class accuracy of VOI v/s random on the scene-13 dataset.

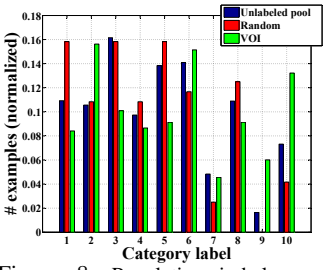
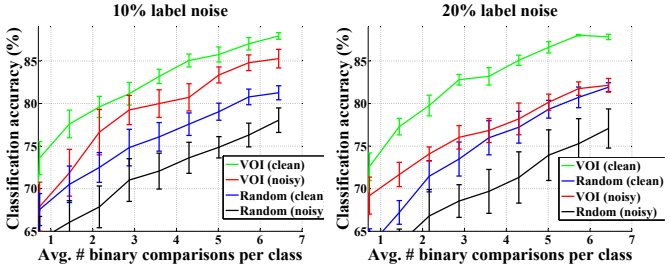



Figure 8. Population imbalance: VOI selects many images even for classes with small populations (see text for details).




(a) (b)

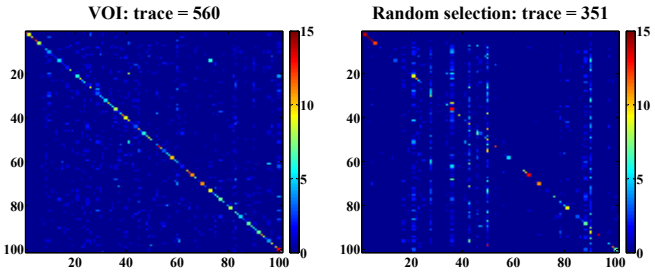
Figure 9. Sensitivity to label noise, (a) 10%, (b) 20%. VOI with noisy data outperforms the random selection with clean data.

### 4.3. Active selection (VOI) v/s random selection

Figure 6 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 lesser confusion, also indicated by the trace of the two matrices. This demonstrates that the algorithm offers large advantages for many category problems. Figure 7 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.

### 4.4. 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




(a) (b)

Figure 6. Confusion matrices with (a) active (VOI), and (b) random selection (max. trace = 1515). VOI leads to much lower confusion.

methods. We artificially impart stochastic labeling noise to the training images. For example, 5% 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 Figure 9.

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

### 4.5. Population imbalance

Real-world data often exhibits class population imbalance, with vastly varying number of examples belonging different classes [6]. 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 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. Figure 8 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 is 0.036 and 0.058 respectively. The significantly smaller deviation shows that active selection overcomes population imbalance to a large extent.

### 4.6. Fast initiation of new classes

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

Table 3. User training time required to encounter all 101 classes.

In Section 2.3.1, we described our method of quickly initiating new classes and then merging the erroneous ones using agglomerative clustering and user feedback. Table 3 summarizes the advantages of the approach (i.e., w/ 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 the 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.

## 5. Conclusions and future work

In this paper, we presented a new multi-class active learning framework that requires only binary feedback from the user. Experiments on large datasets demonstrated the benefits of our approach, in terms of substantially reducing user training time and effort. The proposed method was also shown to be robust to real-world issues such as population imbalance and noise. Future work will focus on choosing even sample images from unlabeled data, in the hope of further reducing training effort.

## 6. Acknowledgment

This work was supported in part by the National Science Foundation through grants #IIS-0219863, #IIP-0443945, #IIP-0726109, #CNS-0708344, and #IIP-0934327, the Minnesota Department of Transportation, and the ITS Institute at the University of Minnesota. We would also like to thank Prof. Kristen Grauman for providing kernel matrices for Caltech-101 data.

## References

[1] A. Asuncion and D. J. Newman. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, 2007. Available at http://archive.ics.uci.edu/ml/datasets.html. 6

[2] S. Basu, A. Banerjee, and R. Mooney. Semi-supervised clustering by seeding. In *ICML*, 2002. 3

[3] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 6

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[5] 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. In *ICDM*, 2001. 5

[6] S. Ertekin, J. Huang, L. Bottou, and L. Giles. Learning on the border: active learning in imbalanced data classification. In *CIKM*, 2007. 7

[7] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005. 6

[8] L. Fei-Fei, P. Perona, and R. Fergus. One-shot learning of object categories. *IEEE Trans. PAMI*, 28(4):594–611, 2006. 2

[9] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *ICCV*, 2005. 6

[10] Y. Guo and R. Greiner. Optimistic active learning using mutual information. In *IJCAI*, 2007. 2, 4, 5

[11] A. Holub, P. Perona, and M. Burl. Entropy-based active learning for object recognition. In *CVPR, Workshop on Online Learning for Classification*, 2008. 1, 6

[12] P. Jain and A. Kapoor. Active learning for large multi-class problems. In *CVPR*, 2009. 1

[13] A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *CVPR*, 2009. 1, 5, 6

[14] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with Gaussian Processes for object categorization. In *ICCV*, 2007. 1, 6

[15] A. Kapoor, E. Horvitz, and S. Basu. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *IJCAI*, 2007. 3

[16] A. Krause and C. Guestrin. Near-optimal nonmyopic value of information in graphical models. In *UAI*, 2005. 3

[17] H.-T. Lin, C.-J. Lin, and R. C. Weng. A note on Platt's probabilistic outputs for support vector machines. *Machine Learning*, 68:267–276, 2007. 2

[18] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001. 6

[19] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 2000. 2

[20] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. 5

[21] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *JMLR*, 2:45–66, 2001. 1

[22] A. Torralba, R. Fergus, and W. T. Freeman. 80 Million tiny images: a large database for non-parametric object and scene recognition. *IEEE Trans. PAMI*, 30:1958–1970, 2008. 1

[23] S. Vijayanarasimhan and K. Grauman. What's it going to cost you? : Predicting effort vs. informativeness for multi-label image annotations. In *CVPR*, 2009. 1, 3

[24] T.-F. Wu, C.-J. Lin, and R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *JMLR*, 5:975–1005, 2004. 2

[25] R. Yan, J. Yang, and A. Hauptmann. Automatically labeling video data using multi-class active learning. In *ICCV*, pages 516–523, 2003. 1