Ninth International Conference on Computer Vision (ICCV'03), Nice, France, October 13-16, 2003, pp. 516-523.

Automatically Labeling Video Data Using Multi-class Active Learning

Rong Yan, Jie Yang, Alexander Hauptmann

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, 15213
{yanrong, yang+, alex+}@cs.cmu.edu

Abstract

Labeling video data is an essential prerequisite for many vision applications that depend on training data, such as visual information retrieval, object recognition, and human activity modeling. However, manually creating labels is not only time-consuming but also subject to human errors, and eventually, becomes impossible for a very large amount of data (e.g. 24/7 surveillance video). To minimize the human effort in labeling, we propose a unified multi-class active learning approach for automatically labeling video data. The contributions of this paper include extending active learning from binary classes to multiple classes and evaluating several practical sample selection strategies. The experimental results show that the proposed approach works effectively even with a significantly reduced amount of labeled data. The best sample selection strategy can achieve more than a 50% error reduction over random sample selection.

1. Introduction

The explosive growth of video sources has created new challenges for the computer vision community. Many applications in computer vision, such as visual information retrieval [1], object recognition [2, 3], and human activity modeling [4], require labeling/annotating of video data. Manually labeling video data, however, is not only a labor intensive and time-consuming task, but also subject to human errors. While much research has been focused on accurate modeling and recognition from video, little attention has been paid to labeling video data efficiently and robustly. The goal of this research is to address the problem of manual video data labeling by developing automated labeling methods within an active learning framework.

In order to get a sense of the difficulties in manually labeling video data, let us consider a problem in geriatric care. Studies indicate that nearly 90% of patients with dementia may exhibit observable agitation, behavior that can be classified as disturbed (a psychiatric or medical condition requiring pharmacological intervention) or disturbing (socially inappropriate behavior that may just be a means of expressing a need). To interpret behavior appropriately or assess why particular behaviors occur, we can use video cameras to monitor patient activities and then analyze video data. Figure 1 depicts examples of such video data taken from multiple locations in a geriatric care center. In order to enable the geriatric care specialist to address situations more accurately and intervene appropriately, we need to create an analysis system that extracts the required data and highlights behaviors of interest. For example, if a specialist would like to observe the behavior of patient A, he/she needs to extract all the frames associated with that patient. To do this, the specialist needs a labeled video sequence where all people have been identified with a label. Suppose we record video at 30 frames/second. For only one camera, we would obtain 259200 frames/day. Manually labeling these data on a frame by frame basis is virtually impossible.

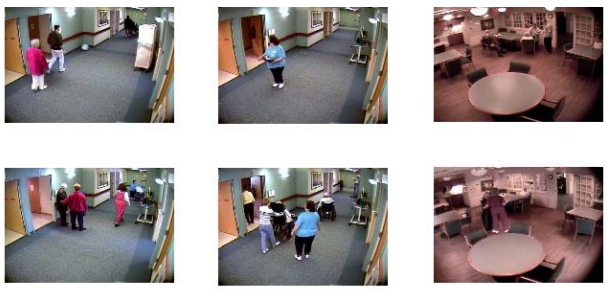

Figure 1. Examples of video data captured from a geriatric care center, where we need to identify different people.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

These difficulties encourage us to develop methods to semi-automatically label video data with less human effort. In fact, many techniques can help to reduce the amount of data to be labeled. Consider the task of labeling people based on identity in the geriatric care video shown in figure 1. Many cues, such as face [2], voice [5], and gait [3] can be used to automatically identify people, although not all the modalities are feasible for this data. Among them, color appearance is the only robust cue for identifying people [6] in this application. This transforms the problem into one of building color appearance models and automatically labeling people using these models. However, we still need some labeled data for training these models. One possible solution is to ask a human to label some randomly selected data, and automatically propagate the labels to the entire collection using a supervised learning algorithm. A better approach would be to select non-random examples which, if labeled, will provide the most information for the learning algorithm. To determine when we stop labeling, we find that it is a subjective judgment with respect to each application as to what accuracy is sufficient and how many labeled examples are needed to reach this accuracy. This motivated us to develop and adapt an incremental learning framework with interaction/supervision from a human. This framework is known as active learning.

In this paper, we propose a unified multi-class active learning approach to minimize human effort in labeling video data. One of our contributions is that we extend the active learning approach from binary classes to multiple classes, which enables the learning algorithm to select the most informative unlabeled data for all the classes instead of just the binary classes. As well as providing a theoretically optimal criterion, we also propose and evaluate several practical sample selection strategies. The experimental results indicate that our multi-class active learning approach works effectively even when the amount of labeled data is significantly reduced. The best sample selection strategies can yield a more than 50% error reduction over random sampling.

2. Problem Description

Typically, the task of automatically labeling video data is to associate unlabeled examples with a single label. For instance, in the task of labeling people's identity, the regions extracted by a people tracker can be thought of as the unlabeled examples, and the people's identities are the labels. This task can be formulated as a multi-class classification problem, where each example is associated with one of the given classes. Let $\mathcal{X}$ denote the domain of possible examples, $\mathcal{Y}$ be a finite set of classes and k be the size of $\mathcal{Y}$. Formally, the learning algorithm takes a set of training examples $(x_1, y_1), ..., (x_m, y_m)$ as input, where $y_i \in \mathcal{Y}$ is the label assigned to example $x_i \in \mathcal{X}$. Typically, the goal of classification is to produce a real-valued hypothesis $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbf{R}$ where f belongs to some hypothesis space $\mathbf{F}$.

The effectiveness of active learning to reduce labeling cost has been demonstrated by previous work [7, 8, 9, 10]. Active learning, or called pool-based active learning, is an interactive learning approach in which the learner has the freedom to select which unlabeled examples should be added into the training set. An active learner may begin with a pool of unlabeled data, select a set of unlabeled examples to be manually labeled as positive or negative and learn from the newly obtained knowledge repetitively. This type of problem can also be called "query learning" [7]. Formally, an active learner l has three major components (f, s, D) [10] with an unlabeled pool $\mathcal{P}$. The first component is the classifier $f(x)$, trained on the labeled data set D. The second one $s_D(\mathcal{P})$ is the sample selection function that selects the most informative examples in the pool $\mathcal{P}$ given the training data D. Compared with a supervised learning approach, the additional component of active learning is the sampling function $s_D(\mathcal{P})$. One of the major tasks for active learning is to determine the sample selection function, which will be discussed in the following section.

3. Multi-Class Active Learning

We present a unified multi-class active learning framework in this section. In the following discussion, we pay particular attention to learning algorithms that attempt to minimize a margin-based loss function, called margin-based learning algorithms [11]. This includes a large family of well-studied algorithms with different loss functions and minimization algorithms, such as decision trees, logistic regression, support vector machines(SVM) and AdaBoost. The margin-based learning algorithms always minimize the loss function with respect to the margin, which is

$$\frac{1}{m}\sum_{i=1}^{m} L(y_i f(x_i)), \tag{1}$$

where L is some loss function $L : \mathbf{R} \rightarrow [0, \infty)$. Without particularly concerning specific learning algorithms, the generalized loss function representation allows us to present more general results as discussed below.

3.1. Output Coding for Multi-Class Classification

Since most margin-based learning algorithms were originally devised for binary classification, the challenge is to extend them to multi-class classification. Several solutions have been proposed to address this issue, in which each class is compared against all others, or in which all pairs of

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

classes are compared to each other, or in which the output is handled with error-correcting output coding(ECOC) [12]. To generalize these approaches, Allwein et al. [11] have proposed a unified approach for decomposing multi-class problems into a set of binary-class problems. To illustrate, we can represent each decomposition by a coding matrix $M \in \{-1, 0, +1\}^{k \times l}$, where k is the number of classes and l is the number of binary classification problems. $M_{rs} = 1$ indicates that the examples in the class r are considered as positive examples for classification problem s. Similarly, for classification problem s, if $M_{rs} = -1$ the examples in the class r are considered negative example, and if $M_{rs} = 0$ we don't care how the learner categorizes the example in the class r. For instance, for one-against-all approach, M is a $k \times k$ matrix with diagonal elements 1 and others -1.

Orthogonal to the problem of coding matrix selection, the learning algorithm has to assign examples to a predicted class $\hat{y} = 1, ..., k$ given the labels provided by binary classifiers. Allwein et al. [11] suggest two types of coding schemes to fuse the predictions: Hamming decoding and loss-based decoding. Let M_r be row r of M and let $f(x)$ be the predictions for an example x for multiple binary learning algorithms, $f(x) = (f_1(x), ..., f_l(x))$. Therefore, the distance measure of Hamming decoding is

$$d_H(M_r, f(x)) = \sum_{s=1}^{l} \left(\frac{1 - sign(m_{rs} f_s(x)}{2} \right). \quad (2)$$

To take advantage of the confidence of binary predictions, they also proposed a loss-based decoding scheme

$$d_L(M_r, f(x)) = \sum_{s=1}^{l} L\left(m_{rs} f_s(x)\right), \quad (3)$$

where L is a loss function for both decoding schemes. The predicted label $\hat{y}$, therefore, is computed as

$$\hat{y} = arg \min_r d(M_r, f(x)).$$

Hamming decoding has been shown to be less effective in classification than loss-based decoding, and thus we use loss-based decoding to combine the predictions of binary classfiers. Several choices of the loss function are possible and it is not clear which works best. Allwein et al. [11] proposed to set L as the same loss function used by the learning algorithm, which is the strategy we adopted in our experiments.

3.2. Optimal Multi-Class Active Learning

As mentioned before, one of the most important components in active learning is the sample selection function which selects a set of informative examples to label. The optimal active learner is an active learner that always asks for labels of those unlabeled examples which, once incorporated into training set, will lead to the lowest expected generalization error. However, for a margin-based learning algorithm, it is better to optimize the loss functions directly [11]. Therefore our goal is to search for the unlabeled examples which can minimize the expected loss on the data set.

Let $P(y|x)$ be the conditional distribution over an example x, and $P(x)$ be the marginal distribution of x. The learner has been given a labeled training set D, and output a set of estimated loss $d_L(M_y, f^D(x))$ for every x in the pool $\mathcal{P}$. We denote $d_L(M_y, f^D(x))$ as $d_L(f^D)$ for the sake of simplicity in the following discussion. We can then write the expected risk function of the learner as follows,

$$R(f^D) = E_x E_{y|x}(d_L(f^D)) = \int_{\mathcal{X}} \sum_{y \in \mathcal{Y}} d_L(f^D) P(y|x) P(x) dx.$$

A multi-class active learner has to select a set of unlabeled examples, or query set D^+ from the pool and ask a human for their labels. After every example x_i^* in D^+ is given labels $y_i^* \in \mathcal{Y}$ and added in the training set, an updated learner will be trained on the training set $D^* = D \vee D^+$. The optimal learner can choose the optimal query set D_{opt}^+ so that the updated learner should have the lowest risk,

$$D_{opt}^+ = arg \min_{D^+} R(f^{D^*}) = arg \min_{D^+} R(f^{D \vee D^+})$$

or the largest risk reduction,

$$D_{opt}^+ = arg \max_{D^+} \left(R(f^D) - R(f^{D^*}) \right). \quad (4)$$

Because it is rather difficult to estimate the expected risk over the full distribution, $P(x)$, it is more feasible to measure the risk over all the examples in the pool. Therefore, for the expected risk before selection we have

$$R(f^D) = \frac{1}{|\mathcal{P}|} \sum_{x \in \mathcal{P}} E_{y|x}(d_L(f^D)) \quad (5)$$

and for the expected risk after selection

$$R(f^{D^*}) = \frac{1}{|\mathcal{P}|} \sum_{x \in \mathcal{P}} E_{y|x}(d_L(f^{D^*})). \quad (6)$$

By substituting the equation (5) and (6) into (4), the minimization function becomes

$$R(f^D) - R(f^{D^*}) = \sum_{x \in \mathcal{P}} E_{y|x}(d_L(f^D) - d_L(f^{D^*})). \quad (7)$$

In theory, the maximization of (7) straightforwardly leads to the optimal query set D^+. Unfortunately, in practice, it is intractable to compute all the $2^{|\mathcal{P}|-|\mathcal{D}|}$ possible combinations even in one round, not to mention selecting unlabeled examples iteratively. One of the feasible solutions is to select only one unlabeled example each

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

time, such that the choice of examples greatly reduces to $|\mathcal{P}| - |\mathcal{D}|$. Plus, many learning algorithms like SVM and Naive Bayes have efficient algorithms for incremental learning [13]. Some recent work has demonstrated how to efficiently optimize the expected loss function in the active learning paradigm [14]. However, in this paper we will not go into further details on this direction.

3.3. Approximated Sample Selection Strategies

In this section, we describe several practical sample selection strategies as alternatives for optimal multi-class active learning. As noted before, our task is to infer a new set of examples D^+ to add into the training set, which minimizes the expected generalization risk. It is computationally intensive to maximize (7) by re-learning the classifiers to estimate the new expected risk. We do not want to go through all the possible combinations for D^+ before picking out one query set. Moreover, because typically only a small number of labeled data are available for training, the estimation for $P(y|x)$ might be unreliable. To make multi-class classification more practical, we will use some simple heuristics to simplify our selection strategies.

To re-learn the classifiers for each possible data combination in the pool becomes one of the major computational burdens in maximizing (7). Therefore, our first goal of the approximation is to eliminate the components that have to be reestimated after adding additional data, which is the prediction function $f(D^*)$. To this end, we seek to reduce the computational cost based on two approximations as follows.

Starting from (7) and substituting the $\mathcal{P}$ with D^+, we obtain

$$D^+_{opt} = arg\max_{D^+} \sum_{x\in D^+} E_{y|x}(d_L(f^D) - d_L(f^{D^*})). \quad (8)$$

An intuitive explanation for this approximation is that if an optimal data set D^+ can be found to maximize (8), it can always pick out the most ambiguous examples in the pool, and thus yield the largest expected risk reduction over the entire collection. A similar assumption proposed by [7] suggests that all expected losses for any x in $\mathcal{P}\backslash D^+$ have an equal influence.

Next, if approximating the difference $d_L(f^D) - d_L(f^{D^*})$ in the prediction of $d_L(f^D)$, we have

$$D^+_{opt} = arg\max_{D^+} \sum_{x\in D^+} E_{y|x}(d_L(f^D)). \quad (9)$$

This approximation is based on the observation that the learning algorithm trained on D^* can always provide the correct prediction for any $x \in D^+$. The reasons are twofold. First, in the active learning problem, training data is always sparse especially at the initial stage, while the feature space often has a large number of dimensions. Second, for some kernel machines like SVMs, it is possible to modify the kernel function such that the training data in the new feature space can be linearly separated [10]. Therefore, for every $x \in D^+$, $E_{y|x}d_L(f^{D^*})$ is much smaller than $E_{y|x}d_L(f^D)$, which makes it reasonable to ignore the $d_L(f^{D^*})$ in (8).

So far, the components that have to be relearned frequently are completely removed in (9). To further simplify, we can rewrite (9) into the following equation which maximizes the expected risk for only one example,

$$D^+ = arg\max_x E_{y|x}(d_L(f^D)). \quad (10)$$

Substituting (3) into $E_{y|x}(d_L(f^D))$ in (10), we get

$$\sum_{y\in\mathcal{Y}} \sum_{s=1}^{l} P(y|x)L(m_{ys}f^D_s(x)) \quad (11)$$

$$= \sum_{t=-1,0,1} \sum_{s=1}^{l} P(m_{ys}=t|x)L(tf^D_s(x)), \quad (12)$$

where $P(m_{ys}=t|x)$ stands for $\sum_{y\in\mathcal{Y},m_{ys}=t} P(y|x)$. Note that $P(m_{ys}=0)=0$ for the coding matrices that do not have the element 0, such as the one-against-all coding matrix.

Finally, we come to the issue of providing a better probability estimation for the conditional probability $P(y|x)$. Since the estimation of the true distribution $P(y|x)$ depends on labeled examples, we will estimate it using the training set D. However, the classification confidence presented in the loss function is not the posterior probability, so we can not straightforwardly treat the confidence as the estimation of $P(y|x)$. To compute (12), it is necessary to normalize the output of confidence into the output of the posterior probability. Two probability estimation models can be suggested:

Uniform Guess In this case, the class-conditional probabilities are assumed to be completely unrelated to the labels on the data, that is, $P(m_{ys}=1|x) = P(m_{ys}=-1|x)$ for all the examples. In the case of $P(m_{ys}=0|x)=0$, the probabilities $P(m_{ys}=\pm 1|x)$ are fixed to the constant $1/2$. Therefore in this case, the probabilities $P(m_{ys}|x)$ can be removed from (11), which then becomes

$$arg\max_x \sum_{t=-1,1} \sum_{s=1}^{l} L(tf^D_s(x)). \quad (13)$$

However, the assumption of uniformness may only work at the initial stage of training, where the estimation only has little to do with the actual labels. But later when the classification confidence of $f_s(x_i)$ is expected to be positively correlated to the label y_i, this uniform guess assumption will always fail.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

Best Worst Case The best worse case model has been suggested in several papers [9, 7]. This model approximates the expected loss function with the smallest loss function among all the possible labels. It implies that the loss function can be expected to be small since the most confident labelling will be most likely to be correct. Thus (11) can be rewritten as

$$arg \max_x \min_{y \in \mathcal{Y}} \sum_{s=1}^{l} L(m_{ys} f_s(x)) \tag{14}$$

or let y_x be the predicted label for example x,

$$arg \max_x \sum_{s=1}^{l} L(m_{y_x s} f_s(x)). \tag{15}$$

The reasoning for this model is to choose the most ambiguous examples with the maximum expected loss for the predicted label. In the case of $l = 1$ where only binary classes are predicted, this strategy can be reduced to the problem of choosing the example x with $\max_x L(y, f(x))$. This can be interpreted as selecting the examples closest to the decision boundary, which is a common sample selection criterion in binary active learning tasks [10].

4. Related Work

One of the first statistical analyses was proposed by Cohn et al. [8]. They demonstrated how to construct optimal queries by minimizing the active learner's variance. But it turns out to be difficult to compute a closed form solution of the expected variance for most complicated learners. Roy et al. [14] presented an optimal active learning algorithm by directly minimizing the learner's expected error on the new test examples. A set of computational optimization techniques makes the algorithm much more practical.

Apart from estimating the expected generalization error, another widely used method for the active learning is to optimize a different, non-optimal criterion. Query-by-Committee (QBC) [15] chooses the instances to be labeled that have maximal disagreement among the individual classifiers. In order to achieve the highest information gain, the examples selected by this approach split the version space into two parts of equal size. More recent work [10] extends the QBC approach to Support Vector Machines active learning for a text classification task. For each query, they estimate the reduction of the version space size for each unlabeled example in the pool. Similar to QBC, the examples that most reduce the version space size are chosen as the next query.

However, many of these early studies focused on the binary classification problem. Surprisingly, few of them mention selection strategies in the context of multi-class classification. One of the most relevant papers to our approach was done by Tong et.al [16]. Viewing the multi-class classification problem as an extension of the binary case, they propose a simple heuristic to select the next unlabeled example that minimizes the maximum model loss

$$x = arg \min_x \max_y \prod_i Area(\mathcal{V}_{x,y}^{(i)}). \tag{16}$$

By approximating the size of the version space $Area(\mathcal{V}_{x,y}^{(i)})$ with $(1 + y_i f_i(x)) Area(\mathcal{V}^{(i)})/2$, the heuristic becomes

$$arg \max_x \min_y \sum_i \log \left(\frac{1}{1 + y_i f_i(x)} \right), \tag{17}$$

where y_i is 1 if label y for instance x is class i, otherwise -1. By comparing (17) and (14), we can show that this heuristic is a special case of our best worst case model with loss function $\log 1/(1 + x)$.

5. Results

In this section, we describe the experiments on creating labels for identifying people in geriatric video data and demonstrate the effectiveness of our multi-class active learning approach. For the sake of simplicity, we simulated the human labeling process using complete, true data labels without actually asking a human for the labels at each step.

5.1. Experimental Setting

The training data was extracted from a 6 hour long, single day and single view geriatric nursing home video, which was sampled at a resolution of 320×240 and a rate of 30 frames per second. Many people contained in the video are partially occluded and there are large variations of the lighting environment. For testing purposes, the noise from background had to be removed. We automated this process using a background subtraction people tracker, which outputs a set of people images. Every image corresponds to the extracted silhouette of a moving person.

From the images generated by the tracker, we sampled a training image collection which included 11 people/classes of interest. In this experiment, we only considered images that did not have any foreground segments containing two or more people. Also, only images where the size of the foreground was larger than 2% of the screen size were kept in our collection, which is reliable for color histogram computation. Finally, over 1,000 single-person images were collected and manually labeled with one of the 11 given classes.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

Since a color histogram is relatively insensitive to the variations of the target appearance due to viewpoint and occlusion [6], we represent the images using a histogram of various color spaces in the following experiments. Though only color feature is considered in this paper, our approach can also be in principle extended to more features such as texture, shape and gait features, which have their own strength and weakness in the human prediction.

5.2. Multiclass Active Learning Results

Constructing an effective active learner is dependent on several factors, including choosing a well-suited binary classifier, loss function, coding matrix, probability estimation scheme and sample size per run. In this section, we will investigate how each factor can affect the performance of a multi-class active learner.

The first series of experiments was designed to verify the contrastive performance of various classifiers and color spaces. In this experiment, we split the image collection into halves. The first 50% of the image collection were used as training data and the remaining 50% were used as test data. For each image, a color histogram was generated in both the RGB(Red-Green-Blue) and HSV(Hue-Saturation-Value) color spaces [17], where each color channel had a fixed number of 32 bins. Thus we have a total of 96 one-dimensional features for both color spaces. Table 1 shows the misclassification rate for the following classifiers: k Nearest Neighbor with L_2 distance (kNN L_2), k Nearest Neighbor with χ^2 distance(kNN χ^2), Linear SVM (LSVM), Radial Kernel SVM with $\rho = 0.01$ (RSVM) and χ^2 kernel SVM with $\rho = 0.01$ (SVM χ^2) [1]. As anticipated, SVM χ^2 achieves the best performance followed by kNN χ^2, indicating that the χ^2 distance is the best suited distance metric for histograms of color features [17]. Theoretically, HSV space can be considered to be more suitable than RGB space for image classification since it is less sensitive to change in illuminance. But in practice this issue seems to be minor. The performance of the RGB space is close or even superior to the HSV space for some classifiers such as kNNχ^2. Again, this observation is consistent with previous work [17]. In our following experiments, the χ^2 kernel SVM in HSV color space was chosen as our base classifier due to its top performance.

We evaluated the performance for the parameters of the multi-class active learning approach. Initially, our training data consisted of the first 21 images in the image collection, which contained at least one example for each class. For each run, the active learner selected 10 more unlabelled data points to label. The learning process iterated for 3 runs and ended up with 51 training examples. The remaining data in the collection was then tested. We first experimented with the effects of different loss functions and coding matrices. As suggested by Allwein et al. [11], we used $(1-y)_+$ loss function to predict the class labels for SVM classifiers and computed the expected loss functions with (15). The best worst case model was employed to predict the class-conditional probability. We considered the performance of five different types of loss functions: the exponential loss e^{-x} (EXP), the L_1 loss $1/(1+e^{2yf(x)})$ (L_1), the loss function $(1-y)_+$ ($(1-y)_+$) as well as the minimal margin loss function (MinMG) [2] e^{-100x}. As a baseline, a random loss function was also tested where the active learner randomly selected the unlabeled examples to label. In the future, we plan to study the effects of various other coding schemes. In this work, we experimented with four types of coding matrices, that is, one-against-all(1-vs-r), ECOC with 15bit BCH code(BCH15), ECOC with the first 15 bits of 63bit BCH code(BCH63) and pairwise coding.

The results in table 2 indicate that except pairwise coding most of the loss functions can greatly reduce the error rate compared with the baseline, i.e. the random loss function. We conjecture that the worse performance of pairwise

	kNN L_2	kNN χ^2	LSVM	RSVM	SVM χ^2
HSV	0.152	0.142	0.195	0.144	0.107
RGB	0.125	0.109	0.242	0.115	0.103

Table 1. Misclassification rate for different classifiers in RGB/HSV color space with 50%/50% training/testing splits

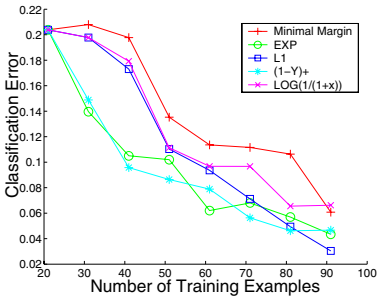

Figure 2. Comparison of classification errors of five different loss functions all using the best worse case model and the one-against-all coding scheme.

[1] k is set to 1 which achieves the best accuracy for any k, $1 \leq k \leq 5$. More details about SVM classifiers and χ^2 kernel can be found at [17]. Note that although some of the classifiers are not margin-based classifiers, we listed them here for the purpose of comparison

[2] We call this minimal margin loss function because $\sum_i e^{-100x_i} \approx e^{-a \min_i x_i}$

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

	MinMG	EXP	L_1	$(1-y)_+$	MinMG	Random
1-vs-r	0.135276	0.101977	0.110302	0.086368	0.111342	0.133195
BCH15	0.096774	0.08949	0.081165	**0.057232**	0.104058	0.122789
BCH63	0.072841	0.099896	0.121748	0.087409	0.090531	0.134235
Pairwise	0.126951	0.186264	0.173777	0.186035	0.164412	0.125099

Table 2. Misclassification rate for different loss functions and coding matrices in multi-class active learning. All training data was limited to a maximum of 50 training examples.

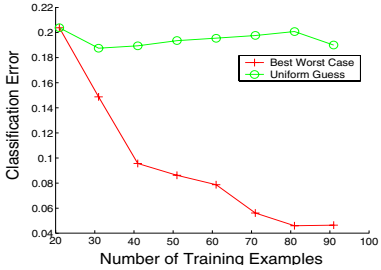

Figure 3. Comparison of classification errors from the best worse case model and the uniform guess model with a $(1-y)_+$ loss function and a one-against-all coding scheme.

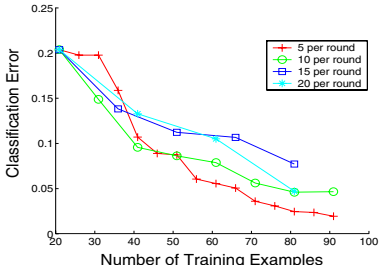

Figure 4. Comparison of classification errors from different sampling sizes per run with the best worse case strategy, $(1-y)_+$ loss function and the one-against-all coding scheme.

coding is due to insufficient training data of each class. The 5.7% error rate obtained by the loss function $(1-y)_+$ and BCH15 coding represent a more than two-fold error reduction. For the coding matrix, BCH15 achieves the lowest error rate, and generally the ECOC coding scheme can reduce classification error by 2%-3% . Taking a closer look at how the active learner behaves when the training data size increases, we plot the curve of the misclassification rate with different loss functions as a function of the training data size in figure 2. The active learner selects 10 unlabeled examples per run iteratively until it reaches 90 training examples. We observe that the classification error always decreases when more training data are added. But the loss function of $(1-y)_+$ and EXP converge faster to their best performance, in other words, the active learner with these two loss functions can achieve better performance with fewer training examples than the others.

To avoid an explosion of loss functions and coding scheme combinations, we restricted the loss function to be $(1-y)_+$ and the coding scheme to one-against-all in the following experiments. Again, we ran the active learner until it had more than 90 training examples. In figure 3, we compare the performance of best worse case model with uniform guess model. Since the uniform guess model cannot pick out informative examples to label, it is unable to improve the classification performance by getting more labels. Surprisingly, its performance is worse than the random loss function in the best worse model. This underscores the importance of a proper probability estimation model.

The last experiment was designed to assess the performance of the sampling size per run. All the active learners began with 21 training examples and stopped at the point of reaching 90 training examples. The classification errors are presented in figure 4 with 5, 10, 15, 20 samples per run. Roughly speaking, lowering the sampling size per run improves the accuracy for the active learner. A partial explanation is that it is more flexible and effective for the active learners to control two rounds of 5 examples than one round of 10 examples.

6. Conclusion

Dealing with vast amounts of unlabeled data is a growing problem in computer vision. In this paper, we proposed a unified multi-class active learning framework in order to reduce the human labeling effort. Our experiments demonstrate that an active learner with careful sample selection can achieve remarkably good performance (5.7% labeling error) with much less human labeling effort (50 examples, which translates into only 5% of the labeling effort) compared to supervised learning. Also, an active learner with

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

the proposed sample selection strategies can do much better than one with only a random sampling strategy, which yields an over two-fold error reduction.

Note that in some cases, the test set might have some data with new labels which do not exist in the training set. This makes the learning problem even harder due to the open label set. We suggest introducing a new "null" label to handle all the objects with unseen labels in the test set and feeding them back to users to label. Another promising avenue for future work is to extend our experiments to multiple day, multiple camera view and multi-person video data with more discriminative features. Moreover, as an alternative of manual labeling, the labels can be obtained from different types of multi-modal information, like face recognition and speaker identification. It would be interesting to study how to fuse these different types of information into a multi-class active learning framework. We plan to explore these extensions in the future.

Acknowledgments

This research is partially supported by the National Science Foundation and the Department of Defense (United States of America) through award numbers 0205219 and N41756-03-C4024. We also thank Rong Jin and Jian Zhang for stimulating discussion.

References

[1] S. Antania, R. Kasturi, and R. Jain, "A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video," *Pattern Recognition*, vol. 4, pp. 945–65, April 2002.

[2] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition'94 (CVPR'94)*, Seattle, WA, June 1994, pp. 568–574.

[3] A. Kale, A. N. Rajagopalan, N. Cuntoor, and V. Kruger, "Gait-based recognition of humans using continuous hmms," in *Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, 2002, pp. 321–326.

[4] C. Bregler and J. Malik, "Learning and recognizing human dynamics in video sequences," in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition'97 (CVPR'97)*, San Juan, PR, 1997, pp. 568–574.

[5] H. Gish and M. Schmidt, "Text-independent speaker identification," *IEEE Signal Processing Magazine*, vol. 11, no. 4, pp. 18–32, 1994.

[6] J. Yang, X. Zhu, R. Gross, J. Kominek, Y. Pan, and A. Waibel, "Multimodal people ID for a multimedia meeting browser," in *ACM Multimedia*, 1999, pp. 159–168.

[7] C. Campbell, N. Cristianini, and A. Smola, "Query learning with large margin classifiers," in *Proc. 17th International Conference on Machine Learning(ICML00)*, 2000, pp. 111–118.

[8] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[9] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *Proc. 17th International Conf. on Machine Learning*. 2000, pp. 839–846, Morgan Kaufmann, San Francisco, CA.

[10] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," in *Proceedings of 17th International Conference on Machine Learning(ICML00)*, 2000, pp. 999–1006.

[11] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," in *Proc. 17th International Conference on Machine Learning(ICML00)*, 2000, pp. 9–16.

[12] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research*, vol. 2, pp. 263–286, 1995.

[13] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advance Neural Information Processing Systems*, 2000, vol. 13, pp. 409–415.

[14] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *Proc. 18th International Conference on Machine Learning(ICML01)*, 2001, pp. 441–448.

[15] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the Fifth Workshop on Computaional Learning Theory*, San Mateo, CA, 1992, pp. 287–294.

[16] S. Tong, *Active Learning: Theory and Applications*, Ph.D. thesis, Stanford University, CA, 2001.

[17] O. Chapelle, P. Haffner, and V. Vapnik, "SVMs for histogram-based image classification," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1055–1065, 1999.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 **IEEE**