A Convex Optimization Framework for Active Learning

Ehsan Elhamifar
University of California, Berkeley

Guillermo Sapiro
Duke University

Allen Yang, S. Shankar Sastry
University of California, Berkeley

Abstract

In many image/video/web classification problems, we have access to a large number of unlabeled samples. However, it is typically expensive and time consuming to obtain labels for the samples. Active learning is the problem of progressively selecting and annotating the most informative unlabeled samples, in order to obtain a high classification performance. Most existing active learning algorithms select only one sample at a time prior to retraining the classifier. Hence, they are computationally expensive and cannot take advantage of parallel labeling systems such as Mechanical Turk. On the other hand, algorithms that allow the selection of multiple samples prior to retraining the classifier, may select samples that have significant information overlap or they involve solving a non-convex optimization. More importantly, the majority of active learning algorithms are developed for a certain classifier type such as SVM. In this paper, we develop an efficient active learning framework based on convex programming, which can select multiple samples at a time for annotation. Unlike the state of the art, our algorithm can be used in conjunction with any type of classifiers, including those of the family of the recently proposed Sparse Representation-based Classification (SRC). We use the two principles of classifier uncertainty and sample diversity in order to guide the optimization program towards selecting the most informative unlabeled samples, which have the least information overlap. Our method can incorporate the data distribution in the selection process by using the appropriate dissimilarity between pairs of samples. We show the effectiveness of our framework in person detection, scene categorization and face recognition on real-world datasets.

1. Introduction

The goal of recognition algorithms is to obtain the highest level of classification accuracy on the data, which can be images, videos, web documents, etc. The common first step of building a recognition system is to provide the machine learner with labeled training samples. Thus, in supervised and semi-supervised frameworks, the classifier's performance highly depends on the quality of the provided labeled training samples. In many problems in computer vision, pattern recognition and information retrieval, it is fairly easy to obtain a large number of unlabeled training samples, e.g., by downloading images, videos or web documents from the Internet. However, it is, in general, difficult to obtain labels for the unlabeled samples, since the labeling process is typically complex, expensive and time consuming. Active learning is the problem of progressively selecting and annotating the most informative data points from the pool of unlabeled samples, in order to obtain a high classification performance.

Prior Work. Active learning has been well studied in the literature with a variety of applications in image/video categorization [5, 15, 16, 22, 30, 33, 34, 37], text/web classification [25, 29, 38], relevance feedback [3, 36], etc. The majority of the literature consider the *single mode* active learning [21, 23, 25, 27, 29, 31], where the algorithm selects and annotates only one unlabeled sample at a time prior to retraining the classifier. While this approach is effective in some applications, it has several drawbacks. First, there is a need to retrain the classifier after adding each new labeled sample to the training set, which can be computationally expensive and time consuming. Second, such methods cannot take advantage of parallel labeling systems such as Mechanical Turk or LabelMe [7, 24, 28], since they request annotation for only one sample at a time. Third, single mode active learning schemes might select and annotate an outlier instead of an informative sample for classification [26]. Fourth, these methods are often developed for a certain type of a classifier such as SVM or Naive Bayes and cannot be easily modified to work with other classifier types [21, 23, 25, 29, 31].

To address some of the above issues, more recent methods have focused on the *batch mode* active learning, where they select and annotate multiple unlabeled samples at a time prior to retraining the classifier [2, 5, 12, 17, 18]. Notice that one can run a single mode active learning method multiple times without retraining the classifier in order to select multiple unlabeled samples. However, the drawback of this approach is that the selected samples can have significant information overlap, hence, they do not improve

Figure 1. We demonstrate the effectiveness of our proposed active learning framework on three problems of person detection, scene categorization and face recognition. Top: sample images from the INRIA Person dataset [6]. The dataset contains images from 2 classes, either containing people or not. Middle: sample images from the Fifteen Scene Categories dataset [19]. The dataset contains images from 15 different categories, such as street, building, mountain, etc. Bottom: sample images from the Extended YaleB Face dataset [20]. The dataset contains images from 38 classes, corresponding to 38 different individuals, captured under a fixed pose and varying illumination.

the classification performance compared to the single mode active learning scheme. Other approaches try to decrease the information overlap among the selected unlabeled samples [2, 12, 13, 18, 36]. However, such methods are often ad-hoc or involve a non-convex optimization, which cannot be solved efficiently [12, 13], hence approximate solutions are sought. Moreover, similar to the single mode active learning, most batch mode active learning algorithms are developed for a certain type of a classifier and cannot be easily modified to work with other classifier types [12, 13, 17, 29, 32, 34].

Paper Contributions. In this paper, we develop an efficient active learning framework based on convex programming that can be used in conjunction with any type of classifiers. We use the two principles of classifier uncertainty and sample diversity in order to guide the optimization program towards selecting the most informative unlabeled samples. More specifically, for each unlabeled sample, we define a confidence score that reflects how uncertain the sample's predicted label is according to the current classifier and how dissimilar the sample is with respect to the labeled training samples. A large value of the confidence score for an unlabeled sample means that the current classifier is more certain about the predicted label of the sample and also the sample is more similar to the labeled training samples. Hence, annotating it does not provide significant additional information to improve the classifier's performance. On the other hand, an unlabeled sample with a small confidence score is more informative and should be labeled. Since we can have many unlabeled samples with low confidence scores and they may have information overlap with each other, i.e., can be similar to each other, we need to select a few representatives of the unlabeled samples with low confidence scores. We perform this task by employing and modifying a recently proposed algorithm for finding data representatives based on simultaneous sparse recovery [9]. The algorithm that we develop has the following advantages with respect to the state of the art:

– It addresses the batch mode active leaning problem, hence, it can take advantage of parallel annotation systems such as Mechanical Turk and LabelMe.

– It can be used in conjunction with any type of classifiers. The choice of the classifier affects selection of unlabeled samples through the confidence scores, but the proposed framework is generic. In fact, in our experiments, we consider the problem of active learning using the recently proposed Sparse Representation-based Classification (SRC) method [35]. To the best of our knowledge, this is the first active learning framework for the SRC algorithm.

– It is based on convex programming, hence can be solved efficiently. Unlike the state of the art, it incorporates both the classifier uncertainty and sample diversity in a convex optimization to select multiple informative samples that are diverse with respect to each other and the labeled samples.

– It can incorporate the distribution of the data by using an appropriate dissimilarity matrix in the convex optimization program. The dissimilarity between pairs of points can be Euclidean distances (when the data come from a mixture of Gaussians), geodesic distances (when data lie on a manifold) or other types of content/application-dependent dissimilarity, which we do not restrict to come from a metric.

Paper Organization. The organization of the paper is as follows. In Section 2, we review the Dissimilarity-based Sparse Representative Selection (DSMRS) algorithm that we leverage upon in this paper. In Section 3, we propose our framework of active learning. We demonstrate experimental results on multiple real-world problems in Section 4. Finally, Section 5 concludes the paper.

2. Dissimilarity-based Sparse Modeling Representative Selection (DSMRS)

In this section, we review the Dissimilarity-based Sparse Modeling Representative Selection (DSMRS) algorithm [9, 10] that finds representative points of a dataset. Assume we have a dataset with N points and we are given dissimilarities $\{d_{ij}\}_{i,j=1,\dots,N}$ between every pair of points. d_{ij} denotes how well i represents j. The smaller the value of d_{ij} is, the better point i is a representative of point j. We assume that the dissimilarities are nonnegative and $d_{jj} \leq d_{ij}$ for every i and j. We can collect the dissimilarities in a matrix as

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{d}_1^\top \\ \vdots \\ \boldsymbol{d}_N^\top \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & d_{NN} \end{bmatrix} \in \mathbb{R}^{N \times N}. \qquad (1)$$

Given the dissimilarities, the goal is to find a few points that well represent the dataset. To do so, [9] proposes a convex optimization framework by introducing variables z_{ij} associated to d_{ij}. $z_{ij} \in [0,1]$ indicates the probability that i is a representative of j. We can collect the optimization variables in a matrix as

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_1^\top \\ \vdots \\ \boldsymbol{z}_N^\top \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ z_{N1} & z_{N2} & \cdots & z_{NN} \end{bmatrix} \in \mathbb{R}^{N \times N}. \qquad (2)$$

In order to select *a few* representatives that *well encode* the collation of points in the dataset, two objective functions should be optimized. The first objective function is the encoding cost of the N data points via the representatives. The encoding cost of j via i is set to $d_{ij} z_{ij} \in [0, d_{ij}]$, hence the total encoding cost for all points is

$$\sum_{i,j} d_{ij} z_{ij} = \operatorname{tr}(\boldsymbol{D}^\top \boldsymbol{Z}). \qquad (3)$$

The second objective function corresponds to penalizing the number of selected representatives. Notice that if i is a representative of some points in the dataset, then $\boldsymbol{z}_i \neq \boldsymbol{0}$ and if i does not represent any point in the dataset, then $\boldsymbol{z}_i = \boldsymbol{0}$. Hence, the number of representatives corresponds to the number of nonzero rows of $\boldsymbol{Z}$. A convex surrogate for the cost associated to the number of selected representative is given by

$$\sum_{i=1}^{N} \|\boldsymbol{z}_i\|_q \triangleq \|\boldsymbol{Z}\|_{1,q}, \qquad (4)$$

where $q \in \{2, \infty\}$. Putting the two objectives together, the DSMRS algorithm solves

$$\min \; \lambda \|\boldsymbol{Z}\|_{q,1} + \operatorname{tr}(\boldsymbol{D}^\top \boldsymbol{Z}) \quad \text{s.t.} \quad \boldsymbol{Z} \geq 0, \; \boldsymbol{1}^\top \boldsymbol{Z} = \boldsymbol{1}^\top, \qquad (5)$$

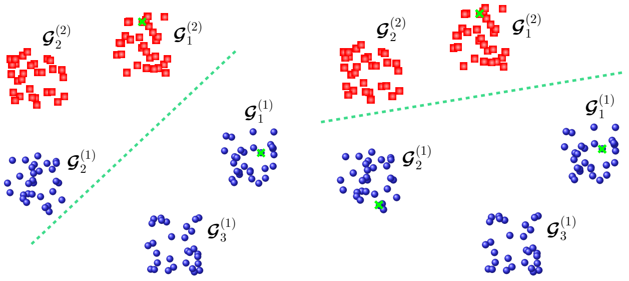

Figure 2. Separating data in two different classes. Class 1 consists of data in $\{\mathcal{G}_1^{(1)}, \mathcal{G}_2^{(1)}, \mathcal{G}_3^{(1)}\}$ and class 2 consists of data in $\{\mathcal{G}_1^{(2)}, \mathcal{G}_2^{(2)}\}$. Left: a max-margin linear SVM learned using two training samples (green crosses). Data in $\mathcal{G}_2^{(1)}$ are misclassified as belonging to class 1. Note that labeling samples from $\mathcal{G}_3^{(1)}$ or $\mathcal{G}_2^{(2)}$ does not change the decision boundary much and $\mathcal{G}_2^{(1)}$ will be still misclassified. Right: labeling a sample that the classifier is more uncertain about its predicted class, helps to improve the classification performance. In this case, labeling a sample from $\mathcal{G}_2^{(1)}$ that is close to the decision boundary, results in changing the decision boundary and correct classification of all samples.

where the constraints ensure that each column of $\boldsymbol{Z}$ is a probability vector, denoting the association probability of j to each one of the data points. Thus, the nonzero rows of the solution $\boldsymbol{Z}$ indicate the indices of the representatives. Notice that $\lambda > 0$ balances the two costs of the encoding and the number of representatives. A smaller value of λ puts more emphasis on better encoding, hence results in obtaining more representatives, while a larger value of λ puts more emphasis on penalizing the number of representatives, hence results in obtaining less representatives.

3. Active Learning via Convex Programming

In this section, we propose an efficient algorithm for active learning that takes advantage of convex programming in order to find the most informative points. Unlike the state of the art, our algorithm can be used in conjunction with any classifier type. To do so, we use the two principles of classifier uncertainty and sample diversity to define confidence scores for unlabeled samples. A lower confidence score for an unlabeled sample indicates that we can obtain more information by annotating that sample. However, the number of unlabeled samples with low confidence scores can be large and, more importantly, the samples can have information overlap with each other or they can be outliers. Thus, we integrate the confidence scores in the DSMRS framework in order to find a few representative unlabeled samples that have low confidence scores. In the subsequent sections, we define the confidence scores and show how to use them in the DSMRS framework in order to find the most informative samples. We assume that we have a total of N samples, where $\mathcal{U}$ and $\mathcal{L}$ denote sets of indices of unlabeled and labeled samples, respectively.

3.1. Classifier Uncertainty

First, we use the classifier uncertainty in order to select informative points for improving the classifier performance. The uncertainty sampling principle [4] states that the informative samples for classification are the ones that the classifier is most uncertain about.

To illustrate this, consider the example shown in the left plot of Figure 2, where the data belong to two different classes. $\mathcal{G}_j^{(i)}$ denotes the j-th cluster of samples that belong to class i. Assume that we already have two labeled samples, shown by green crosses, one from each class. For this specific example, we consider the linear SVM classifier but the argument is general and applies to other classifier types. A max-margin hyperplane learned via SVM for the two training samples is shown in the figure. Notice that the classifier is more confident about the labels of samples in $\mathcal{G}_3^{(1)}$ and $\mathcal{G}_2^{(2)}$ as they are farther from the decision boundary, while it is less confident about the labels of samples in $\mathcal{G}_2^{(1)}$, since they are closer to the hyperplane boundary. In this case, labeling any of the samples in $\mathcal{G}_3^{(1)}$ or $\mathcal{G}_2^{(2)}$ does not change the decision boundary, hence, samples in $\mathcal{G}_2^{(1)}$ will still be misclassified. On the other hand, labeling a sample from $\mathcal{G}_2^{(1)}$ changes the decision boundary so that points in the two classes will be correctly classified, as shown in the right plot of Figure 2.

Now, for a generic classifier, we define its confidence about the predicted label of an unlabeled sample. Consider data in L different classes. For an unlabeled sample i, we consider the probability vector $\boldsymbol{p}_i = \begin{bmatrix} p_{i1} & \cdots & p_{iL} \end{bmatrix}$, where p_{ij} denotes the probability that sample i belongs to class j. We define the *classifier confidence score* of point i as

$$c_{\mathit{classifier}}(i) \triangleq \sigma - (\sigma - 1)\frac{E(\boldsymbol{p}_i)}{\log_2(L)} \in [1, \sigma], \qquad (6)$$

where $\sigma > 1$ and $E(\cdot)$ denotes the entropy function. Note that when the classifier is most certain about the label of a sample i, i.e., only one element of $\boldsymbol{p}_i$ is nonzero and equal to one, then the entropy is zero and the confidence score is maximum, i.e., is equal to σ. On the other hand, when the classifier is most uncertain about the label of a sample i, i.e., when all the elements of $\boldsymbol{p}_i$ are equal to $1/L$, then the entropy is equal to $\log_2(L)$ and the confidence score is minimum, i.e., is equal to one.

Remark 1 *For probabilistic classifiers such as Naive Bayes, the probability vectors, $\boldsymbol{p}_i$, are directly given by the output of the algorithms. For SVM, we use the result of [14] to estimate $\boldsymbol{p}_i$. For SRC, we can compute the multi-class probability vectors as follows. Let $\boldsymbol{x}_i = \begin{bmatrix} \boldsymbol{x}_{i1}^\top & \cdots & \boldsymbol{x}_{iL}^\top \end{bmatrix}^\top$ be the sparse representation of an unlabeled sample i, where $\boldsymbol{x}_{ij}$ denotes the representation coefficients using labeled samples from class j. We set $p_{ij} \triangleq$* $\|\boldsymbol{x}_{ij}\|_1/\|\boldsymbol{x}_i\|_1$.

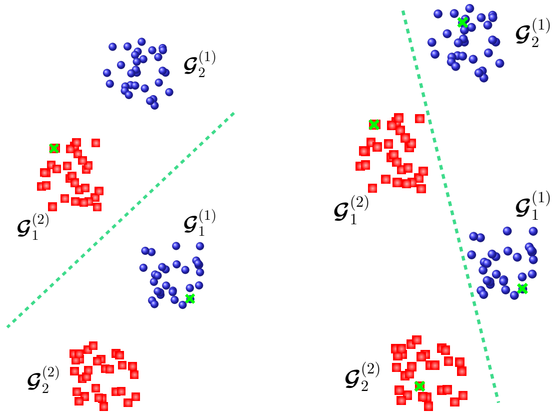

Figure 3. Separating data in two different classes. Class 1 consists of data in $\{\mathcal{G}_1^{(1)}, \mathcal{G}_2^{(1)}\}$ and class 2 consists of data in $\{\mathcal{G}_1^{(2)}, \mathcal{G}_2^{(2)}\}$. Left: a max-margin linear SVM learned using two training samples (green crosses). Data in $\mathcal{G}_2^{(1)}$ and $\mathcal{G}_2^{(2)}$ are misclassified as belonging to class 2 and 1, respectively. Note that the most uncertain samples according to the classifier are samples from $\mathcal{G}_1^{(1)}$ and $\mathcal{G}_1^{(2)}$, which are close to the decision boundary. However, labeling such samples does not change the decision boundary much and samples in $\mathcal{G}_2^{(1)}$ and $\mathcal{G}_2^{(2)}$ will still be misclassified. Right: labeling samples that are sufficiently dissimilar from the labeled training samples helps to improve the classification performance. In this case, labeling a sample from $\mathcal{G}_2^{(1)}$ and a sample from $\mathcal{G}_2^{(2)}$ results in changing the decision boundary and correct classification of all samples.

3.2. Sample Diversity

We also use the sample diversity criterion in order to find the most informative points for improving the classifier performance. More specifically, sample diversity states that informative points for classification are the ones that are sufficiently dissimilar from the labeled training samples (and from themselves in the batch mode setting).

To illustrate this, consider the example of Figure 3, where the data belong to two different classes. $\mathcal{G}_j^{(i)}$ denotes the j-th cluster of samples that belong to class i. Assume that we already have two labeled samples, shown by green crosses, one from each class. For this example, we consider the linear SVM classifier but the argument applies to other classifier types. The max-margin hyperplane learned via SVM for the two training samples is shown in the left plot of Figure 3. Notice that samples in $\mathcal{G}_1^{(1)}$ and $\mathcal{G}_1^{(2)}$ are similar to the labeled samples (have small Euclidean distances to the labeled samples in this example). In fact, labeling any of the samples in $\mathcal{G}_1^{(1)}$ or $\mathcal{G}_1^{(2)}$ does not change the decision boundary much, and the points in $\mathcal{G}_2^{(1)}$ will be still misclassified as belonging to class 2. On the other hand, samples in $\mathcal{G}_2^{(1)}$ and $\mathcal{G}_2^{(2)}$ are more dissimilar from the labeled training samples. In fact, labeling a sample from $\mathcal{G}_2^{(1)}$ or $\mathcal{G}_2^{(2)}$ changes the decision boundary so that points in the

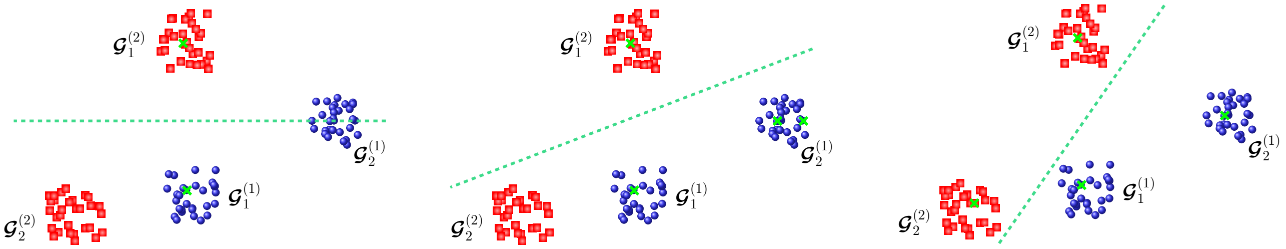

Figure 4. Separating data in two different classes. Class 1 consists of data in $\{\mathcal{G}_1^{(1)}, \mathcal{G}_2^{(1)}\}$ and class 2 consists of data in $\{\mathcal{G}_1^{(2)}, \mathcal{G}_2^{(2)}\}$. Left: a max-margin linear SVM learned using two training samples (green crosses). All samples in $\mathcal{G}_2^{(2)}$ as well as some samples in $\mathcal{G}_2^{(1)}$ are misclassified. Middle: two samples with lowest confidence scores correspond to two samples from $\mathcal{G}_2^{(1)}$ that are close to the decision boundary. A retrained classifier using these two samples, which have information overlap, still misclassifies samples in $\mathcal{G}_2^{(2)}$. Right: two representatives of samples with low confidence scores correspond to a sample from $\mathcal{G}_2^{(1)}$ and a sample from $\mathcal{G}_2^{(2)}$. A retrained classifier using these two samples correctly classifies all the samples in the dataset.

two classes will be correctly classified, as shown in the right plot of Figure 3.

To incorporate diversity with respect to the labeled training set, $\mathcal{L}$, for a point i in the unlabeled set, $\mathcal{U}$, we define the *diversity confidence score* as

$$c_{\textit{diversity}}(i) \triangleq \sigma - (\sigma - 1)\frac{\min_{j\in\mathcal{L}} d_{ji}}{\max_{k\in\mathcal{U}} \min_{j\in\mathcal{L}} d_{jk}} \in [1, \sigma], \tag{7}$$

where $\sigma > 1$. When the closest labeled sample to an unlabeled sample i is very similar to it, i.e., $\min_{j\in\mathcal{L}} d_{ji}$ is close to zero, then the diversity confidence score is large, i.e., is close to σ. This means that sample i does not promote diversity. On the other hand, when all labeled samples are very dissimilar from an unlabeled sample i, i.e., the fraction in (7) is close to one, then the diversity confidence score is small, i.e., is close to one. This means that selecting and annotating sample i promotes diversity with respect to the labeled samples.

3.3. Selecting Informative Samples

Recall that our goal is to have a *batch mode* active learning framework that selects multiple informative and diverse unlabeled samples, with respect to the labeled samples as well as each other, for annotation. One can think of a simple algorithm that selects samples that have the lowest confidence scores. The drawback of this approach is that while the selected unlabeled samples are diverse with respect to the labeled training samples, they can still have significant information overlap with each other. This comes from the fact that the confidence scores only reflect the relationship of each unlabeled sample with respect to the classifier and the labeled training samples and do not capture the relationships among the unlabeled samples.

To illustrate this, consider the example of Figure 4, where the data belong to two different classes. Assume that we already have two labeled samples, shown by green crosses, one from each class. A max-margin hyperplane learned via SVM for the two training samples is shown in the the left plot of Figure 4. In this case, all samples in $\mathcal{G}_2^{(2)}$ as well as some samples in $\mathcal{G}_2^{(1)}$ are misclassified. Notice that samples in $\mathcal{G}_2^{(1)}$ have small classifier and diversity confidence scores and samples in $\mathcal{G}_2^{(2)}$ have small diversity confidence scores. Now, if we select two samples with lowest confidence scores, we will select two samples from $\mathcal{G}_2^{(1)}$, as they are very close to the decision boundary. However, these two samples have information overlap, since they belong to the same cluster. In fact, after adding these two samples to the labeled training set, the retrained classifier, shown in the middle plot of Figure 4, still misclassifies samples in $\mathcal{G}_2^{(2)}$. On the other hand, two representatives of samples with low confidence scores, i.e., two samples that capture the distribution of samples with low confidence scores, correspond to one sample from $\mathcal{G}_2^{(1)}$ and one sample from $\mathcal{G}_2^{(2)}$. As shown in the right plot of Figure 4, the retrained classifier using these two points correctly classifies all of the samples.

To select a few diverse representatives of unlabeled samples that have low confidence scores, we take advantage of the DSMRS algorithm. Let $\boldsymbol{D} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{U}|}$ be the dissimilarity matrix for samples in the unlabeled set $\mathcal{U} = \{i_1, \cdots, i_{|\mathcal{U}|}\}$. We propose to solve the convex program

$$\min\ \lambda \|\boldsymbol{C}\boldsymbol{Z}\|_{1,q} + \operatorname{tr}(\boldsymbol{D}^\top \boldsymbol{Z}) \quad \text{s. t.} \quad \boldsymbol{Z} \geq 0,\ \mathbf{1}^\top \boldsymbol{Z} = \mathbf{1}^\top, \tag{8}$$

over the optimization matrix $\boldsymbol{Z} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{U}|}$. The matrix $\boldsymbol{C} = \operatorname{diag}(c(i_1), \ldots, c(i_{|\mathcal{U}|}))$ is the confidence matrix with the *active learning confidence scores*, $c(i)$, defined as

$$c(i_k) \triangleq \min\{c_{\textit{classifier}}(i_k), c_{\textit{diversity}}(i_k)\} \in [1, \sigma]. \tag{9}$$

More specifically, for an unlabeled sample i_k that has a small confidence score $c(i_k)$, the optimization program puts less penalty on the k-th row of $\boldsymbol{Z}$ being nonzero. On the other hand, for a sample i_k that has a large confidence score $c(i_k)$, the optimization program puts more penalty on the k-th row of $\boldsymbol{Z}$ being nonzero. Hence, the optimization pro-

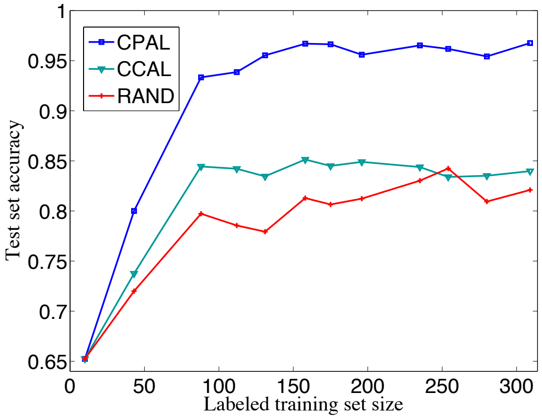

Figure 5. Classification accuracy of different active learning algorithms on the INRIA Person dataset as a function of the total number of labeled training samples selected by each algorithm.

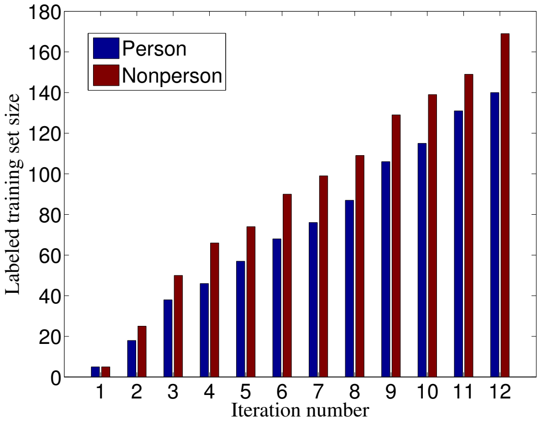

Figure 6. Total number of samples from each class of INRIA Person dataset selected by our proposed algorithm (CPAL) at different active learning iterations.

motes selecting a few unlabeled samples with low confidence scores that are, at the same time, representatives of the distribution of the samples. This therefore addresses the main problems with previous active learning algorithms, which we discussed in Section 1.

Remark 2 *We should note that other combinations of the classifier and diversity scores can be used, such as* $c(i) \triangleq \sqrt{c_{\textit{classifier}}(i) \cdot c_{\textit{diversity}}(i)} \in [1, \sigma]$. *However,* (9) *is very intuitive and works best in our experiments.*

4. Experiments

In this section, we examine the performance of our proposed active learning framework on several real-world applications. We consider person detection, scene categorization and face recognition from real images (see Figure 1). We refer to our approach, formulated in (8), as Convex Programming-based Active Learning (CPAL) and implement it using an Alternating Direction Method of Multipliers method [1], which has quadratic complexity in the number of unlabeled samples. For all the experiments, we fix $\sigma = 20$ in (6) and (7), however, the performances do not change much for $\sigma \in [5, 40]$. As the experimental results show, our algorithm works well with different types of classifiers.

To illustrate the effect of confidence scores and representativeness of samples in the performance of our proposed framework, we consider several methods for comparison. Assume that our algorithm selects K_t samples at iteration t, i.e., prior to training the classifier for the t-th time.

– We select K_t samples uniformly at random from the pool of unlabeled samples. We refer to this method as RAND.

– We select K_t samples that have the smallest classifier confidence scores. For an SVM classifier, this method corresponds to the algorithm proposed in [29]. We refer to this algorithm as Classifier Confidence-based Active Learning (CCAL).

4.1. Person Detection

In this section, we consider the problem of detecting humans in images. To do so, we use the INRIA Person dataset [6] that consists of a set of positive training/test images, which contain people, and a set of negative train/test images, which do not contain a person (see Figure 1). For each image in the dataset, we compute the Histogram of Oriented Gradients (HOG), which has been shown to be an effective descriptor for the task of person detection [6, 8]. We use the positive/negative training images in the dataset to form the pool of unlabeled samples ($2,416$ positive and $2,736$ negative samples) and use the the positive/negative test images for testing ($1,126$ positive and 900 negative samples). For this binary classification problem ($L = 2$), we use the linear SVM classifier, which has been shown to work well with HOG features for the person detection task [6, 8]. We use the χ^2-distance to compute the dissimilarity between the histograms, as it works better than other dissimilarity types, such as the ℓ_1-distance and KL-divergence, in our experiments.

Figure 5 shows the classification accuracy of different active learning methods on the test set as a function of the total number of labeled samples. From the results, we make the following conclusions:

– Our proposed active learning algorithm, consistently outperforms other algorithms. In fact, with 316 labeled samples, CPAL obtains 96% accuracy while other methods obtain less than 84% accuracy on the test set.

– CCAL and RAND perform worse than our proposed algorithm. This comes from the fact that the selected samples by CCAL can have information overlap and are not necessarily representing the distribution of unlabeled samples with

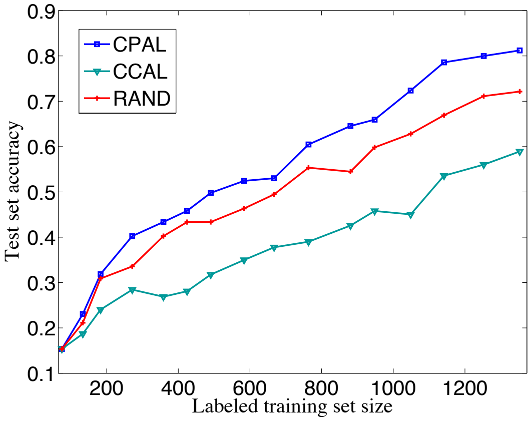

Figure 7. Classification accuracy of different active learning algorithms on the Fifteen Scene Categories dataset as a function of the total number of labeled training samples selected by each algorithm.

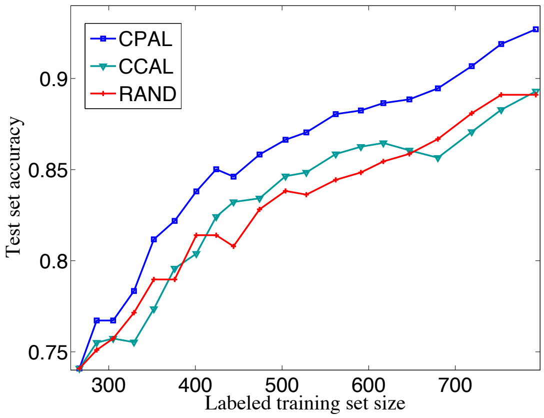

Figure 8. Classification accuracy of different active learning algorithms on the Extended YaleB Face dataset as a function of the total number of labeled training samples selected by each algorithm.

low confidence scores. Also, RAND ignores all confidence scores and obtains, in general, lower classification accuracy than CCAL.

Figure 6 shows the total number of samples selected by our method from each class. Although our active learning algorithm is unaware of the separation of unlabeled samples into classes, it consistently selects about the same number of samples from each class. Notice also that our method selects a bit more samples from the nonperson class, since, as expected, the negative images have more variation than the positive ones.

4.2. Scene Categorization

In this section, we consider the problem of scene categorization in images. We use the Fifteen Scene Categories dataset [19] that consists of images from $L = 15$ different classes, such as coasts, forests, highways, mountains, stores, etc (see Figure 1). There are between 210 and 410 images in each class, making a total of $4,485$ images in the dataset. We randomly select 80% of images in each class to form the pool of unlabeled samples and use the rest of the 20% of images in each class for testing. We use the kernel SVM classifier (one-versus-rest) with the Spatial Pyramid Match (SPM) kernel, which has been shown to be effective for scene categorization [19]. More specifically, the SPM kernel between a pair of images is given by the weighted intersection of the multi-resolution histograms of the images. We use 3 pyramid levels and 200 bins to compute the histograms and the kernel. As the SPM is itself a similarity between pairs of images, we also use it to compute the dissimilarities by negating the similarity matrix and shifting the elements to become non-negative.

Figure 7 shows the accuracy of different active learning methods on the test set as a function of the total number of selected samples. Our method consistently performs better than other approaches. Unlike the experiment in the previous section, here the RAND method, in general, has a better performance than CCAL method that selects multiple samples with low confidence scores. A careful look into the selected samples by different methods shows that, this is due to the fact that CCAL may repeatedly select similar samples from a fixed class while a random strategy, in general, does not get stuck to repeatedly select similar samples from a fixed class.

4.3. Face Recognition

Finally, we consider the problem of active learning for face recognition. We use the Extended YaleB Face dataset [20], that consists of face images of $L = 38$ individuals (classes). Each class consists of 64 images captured under the same pose and varying illumination. We randomly select 80% of images in each class to form the pool of unlabeled samples and use the rest of the 20% of images in each class for testing. We use the Sparse Representation-based Classification (SRC), which has been shown to be effective for the classification of human faces [35]. To the best of our knowledge, our work is the first one addressing the active learning problem in conjunction with SRC. We downsample the images and use the 504-dimensional vectorized images as the feature vectors. We use the Euclidean distance to compute dissimilarities between pairs of samples.

Figure 8 shows the classification accuracy of different active learning methods as a function of the total number of labeled training samples selected by each algorithm. One can see that our proposed algorithm performs better than other methods. With a total of 790 labeled samples (average of 21 samples per class), we obtain the same accuracy (about 97%) as reported in [35] for 32 random samples per class. It is important to note that the performances of RAND and CCAL are very close. This comes from the fact that the space of images from each class are not densely sampled. Hence, samples are typically dissimilar from each other. As a result, samples with low confidence scores are generally dissimilar from each other.

5. Conclusions

We proposed a batch mode active learning algorithm based on simultaneous sparse recovery that can be used in conjunction with any classifier type. The advantage of our algorithm with respect to the state of the art is that it incorporates classifier uncertainty and sample diversity principles via confidence scores in a convex programming scheme. Thus, it selects the most informative unlabeled samples for classification that are sufficiently dissimilar from each other as well as the labeled samples and represent the distribution of the unlabeled samples. We demonstrated the effectiveness of our approach by experiments on person detection, scene categorization and face recognition on real-world images.

Acknowledgment

E. Elhamifar, A. Yang and S. Sastry are supported by grants ONR N000141310341, ONR N000141210609, ONR N000140910230, DARPA FA8650-11-1-7153, ARO 63092-MA-II as well as grants from BAE Systems (94316), DOD Advanced Research Project (30989), Army Research Laboratory (31633), University of Pennsylvania (96045). G. Sapiro acknowledges the support by ONR, NGA, NSF, ARO and AFOSR grants.

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 2010.

[2] K. Brinker. Incorporating diversity in active learning with support vector machines. *ICML*, 2003.

[3] E. Chang, S. Tong, K. Goh, , and C. Chang. Support vector machine concept-dependent active learning for image retrieval. *IEEE Trans. on Multimedia*, 2005.

[4] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Journal of Machine Learning*, 1994.

[5] B. Collins, J. Deng, K. Li, and L. Fei-Fei. Towards scalable dataset construction: An active learning approach. *ECCV*, 2008.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005.

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

[8] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. *CVPR*, 2009.

[9] E. Elhamifar, G. Sapiro, and R. Vidal. Finding exemplars from pairwise dissimilarities via simultaneous sparse recovery. *NIPS*, 2012.

[10] E. Elhamifar and S. S. Sastry. Dissimilarity-based sparse modeling representative selection. *submitted to IEEE Trans. PAMI*, 2013.

[11] R. Greiner, A. Grove, and D. Roth. Learning cost-sensitive active classiers. *Articial Intelligence*, 2002.

[12] Y. Guo and D. Schuurmans. Discriminative batch mode active learning. *NIPS*, 2007.

[13] S. Hoi, R. Jin, J. Zhu, and M. Lyu. Semi-supervised svm batch mode active learning with applications to image retrieval. *ACM Trans. on Information Systems*, 2009.

[14] T. K. Huang, R. C. Weng, and C. J. Lin. Generalized bradley-terry models and multi-class probability estimates. *JMLR*, 2006.

[15] A. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. *IEEE CVPR*, 2009.

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

[17] A. Kovashka, S. Vijayanarasimhan, and K. Grauman. Actively selecting annotations among objects and attributes. *ICCV*, 2011.

[18] A. Krause and C. Guestrin. Nonmyopic active learning of gaussian processes: an exploration-exploitation approach. *ICML*, 2007.

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

[20] K. C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. PAMI*, 2005.

[21] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Journal of Machine Learning*, 2000.

[22] G. J. Qi, X. S. Hua, Y. Rui, J. Tang, and H. J. Zhang. Two-dimensional active learning for image classification. *CVPR*, 2008.

[23] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. *ICML*, 2001.

[24] B. Russell, A. Torralba, K. Murphy, and W. Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 2008.

[25] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. *ICML*, 2000.

[26] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. *Conference on Empirical Methods in Natural Language Processing*, 2008.

[27] B. Siddiquie and A. Gupta. Beyond active noun tagging: Modeling contextual interactions for multi-class active learning. *CVPR*, 2010.

[28] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. *CVPR*, 2008.

[29] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *JMLR*, 2001.

[30] S. Vijayanarasimhan and K. Grauman. Multi-level active prediction of useful image annotations for recognition. *NIPS*, 2008.

[31] S. Vijayanarasimhan and K. Grauman. Whats it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. *CVPR*, 2009.

[32] S. Vijayanarasimhan and K. Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *CVPR*, 2011.

[33] S. Vijayanarasimhan and K. Grauman. Active frame selection for label propagation in videos. *ECCV*, 2012.

[34] S. Vijayanarasimhan, P. Jain, and K. Grauman. Far-sighted active learning on a budget for image and video recognition. *CVPR*, 2010.

[35] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. PAMI*, 2009.

[36] Z. Xu, R. Akella, and Y. Zhang. Incorporating diversity and density in active learning for relevance feedback. *European Conference on Information Retrieval*, 2007.

[37] R. Yan, J. Yang, and A. Hauptmann. Automatically labeling video data using multiclass active learning. *ICCV*, 2003.

[38] C. Zhang and T. Chen. An active learning framework for content-based information retrieval. *IEEE Trans. on Multimedia*, 2002.