Uncertainty Based Active Learning via Sparse Modeling for Image Classification

Gaoang Wang, Jenq-Neng Hwang, Craig Rose, Farron Wallace

***Abstract*—Uncertainty sampling based active learning has been well studied for selecting informative samples to improve the performance of a classifier. In batch mode active learning, a batch of samples are selected for a query at the same time. The samples with top uncertainty are encouraged to be selected. However, this selection strategy ignores the relations among the samples because the selected samples may have much redundant information with each other. This paper addresses this problem by proposing a novel method that combines uncertainty, diversity and density via sparse modeling in the sample selection. We use sparse linear combination to represent the uncertainty of unlabeled pool data with Gaussian kernels, in which the diversity and density are well incorporated. Selective sampling method is proposed before optimization to reduce the representation error. To deal with l_0 norm constraint in the sparse problem, two approximated approaches are adopted for efficient optimization. Four image classification datasets are used for evaluation. Extensive experiments related to batch size, feature space, seed size, significant analysis, data transform and time efficiency demonstrate the advantages of the proposed method.**

***Index Terms*—active learning, sparse modeling, diversity, CNN**

I. Introduction

In real-world applications based on machine learning techniques, it is usually very easy to collect a huge amount of unlabeled data. On the other hand, large number of labeled data are expensive to obtain. In such cases, there would be a huge labeling cost for supervised based learning. Besides that, the classifiers of supervised learning methods are always trained on a specific dataset, and the performance degrades when tested on a slightly different dataset. This is because the testing dataset may not be well represented by the training dataset. Moreover, for practical applications, it is unreasonable to re-train a supervised classifier based on the new dataset. Therefore, we always need to label the new dataset, which is expensive and non-trivial for automatic classification. Fortunately, such problems can be addressed by semi-supervised learning and active learning methods.

Semi-supervised learning methods usually look for additional constraints and the data structures in the unlabeled dataset to improve the performance of trained classifiers [1,2,3,4,34]. In [1], pairwise must-link and cannot-link are taken as constraints for mixture modeling. For image classification, key words associated with both labeled and unlabeled data are used to improve the performance of the semi-supervised classifiers [2]. Manifold regularization for multi-label image classification is taken advantage of in [3]. The transductive support vector machine (TSVM) is also adopted in semi-supervised learning [4]. Some methods for semi-supervised learning are not intrinsically geared to learning from both unlabeled and labeled data, but instead they make use of unlabeled data within a supervised learning framework. Take self-training for example [17,18], a supervised learning algorithm is first trained based on the labeled data. This classifier is then applied to the unlabeled data to generate more labeled examples as input for the supervised learning algorithm. Since the generated labels are not the actual ground truth, errors may be introduced in the training if the initial classifier is not robustly trained. For semi-supervised learning, the training is not stable, and can even collapse if the assumptions and the additional constraints are not actually the fact.

Different from semi-supervised learning, active learning algorithms are able to interactively query the reliable labeler for ground truth to obtain new training data, and eventually overcome the deficiency of semi-supervised learning. Generally, there are two different settings to do the sample selection in active learning. One is purely relying on unsupervised approach to select samples based on the data structure of unlabeled samples without any knowledge of the ground truth labels [14,15,19,20,37]; the other is selecting samples with the help of an initially trained supervised classifier based on a seed set of limited labeled samples [5,6,7,8,35,36,38]. For the first category, since no information of ground truth is given at the beginning, most sample selection strategies are reconstruction based approaches, i.e., the top most informative samples that can represent the whole unlabeled dataset are selected. For the second category, since ground truth labels are provided by a limited number of seed set, the information given by the initially trained classifier can be well utilized. Since the data structure of unlabeled data can also be exploited for the second category, a combination of utilizing the data structure and the initially trained classifier is adopted in recent active learning studies [9,12,13,53,59,61]. In most of such approaches, not only the samples with high uncertainty, but also the samples with representativeness are taken into consideration in the sample selection process.

In addition, different active learning approaches are designed for different applications [43,44,45,46,47,48], respectively. More specifically, active learning is combined with self-paced learning [43] for face identification using convolutional neural networks. Active learning strategy is also explored [44] for training relative attribute ranking functions, with the goal of

requesting human comparisons only where they are most informative. In [45,47], active learning is adopted for image classification problems. Specifically, most informative samples are selected for human labeling based on the output of deep neural networks [45], while in [47] visual and textual information are effectively combined for classification. In [48], a novel approach is proposed for live learning of object detectors, in which the system autonomously refines its models by actively requesting crowd-sourced annotations on images crawled from the Web.

In this paper, we present a novel batch mode approach that combines the information given by an initially trained classifier and the data structure of unlabeled samples via sparse modeling based on uncertainty sampling. We discuss the contributions and advantages of our proposed method as follows.

(1) Represent sample uncertainty via Gaussian kernels. In the sample selection, we use sparse linear combination of Gaussian kernels to represent the uncertainty scores of unlabeled samples. As a result, uncertainty, diversity and density are combined in the sample selection via sparse representation.

(2) Selective sampling. Inspired from [62], we propose selective sampling approach before the optimization. The samples with low uncertainty are filtered out by locality thresholding. There are two advantages of this selective sampling strategy. On one side, the sparse representation is no longer influenced by low uncertainty samples. As a result, the representation error can be largely reduced. On the other side, the number of candidate samples for selection is reduced dramatically, which results in faster convergence during optimization.

(3) Efficient optimization by approximated approaches. We propose two approximated approaches to solve the sparse modeling problem. The first one is based on a greedy search method. Samples are sequentially selected to maximize the reduction of total uncertainty. For the second approach, the sparse representation problem is converted into a quadratic programming formulation.

The outline of the paper is as follows: In Section II, we review some related work of active learning. The sparse modeling of the proposed method is then introduced in Section III. In Section IV, modification of sparse modeling is introduced in the sample selection. Experiments are presented in Section V. Finally, we provide some conclusions and future work in Section VI.

II. Related Work

Active learning shows great power of improving the robustness of classifiers when dealing with limited training data or even without any labeled ground truth. Representativeness of data has been studied in the sample selection strategy when no ground truth labels are given. As it is important to exploit the data distribution when selecting the data to be labeled [16], representativeness sampling tries to select the most representative data points according to the distribution of unlabeled data. For example, some well-known approaches of representativeness sampling [14,15,19,20,37] have been reported. In [20], a simple concept, called transductive experimental design, is proposed to explore available unlabeled data. In [14], the most representative points to reconstruct the whole dataset are selected in active learning by the locally linear reconstruction algorithm. Similarly, in [15], sparsity is taken into consideration in the reconstruction scheme for the sample selection. More recently, locality information by neighborhood samples is utilized in the reconstruction in [19]. However, for representativeness sampling based active learning, since no ground truth label information is given in the experiment setting, the sample selection is purely processed in an unsupervised way. Therefore, the sampling strategy may become inefficient if some assumptions are not met in the unsupervised learning.

On the other hand, some active learning methods take advantage of a set of seed labeled samples to initialize the classifiers [41], such as uncertainty sampling, query-by-committee, expected model change and expected error reduction, etc. Uncertainty sampling is a good way to utilize a pre-trained model in the sample selection. For example, for binary problems, feature points that are close to the classification boundary are chosen to label as the most uncertain samples [5,6,7,8] based on different types of classifiers, like neural networks [5,6] and support vector machines (SVMs) [7,8,35,38,57]. For multi-class classification problems, the first two most likely predictions are used to calculate the uncertainty [10,11]. However, the performance of such uncertainty sampling based active learning largely depends on the robustness of the pre-trained classifiers. Sometimes uncertainty sampling even works worse than random sampling in scenarios when very limited labeled data are used to train the initial classifier [21,22,23].

Since the representativeness of the unlabeled data can also be utilized with a pre-trained classifier, many recent approaches have incorporated the representativeness in their uncertainty design to overcome the weakness of uncertainty sampling based methods [9,12,13,42,53]. In [9], the distribution of the data is taken into consideration in the sample selection. Diversity is incorporated in the version space reduction in [12]. In [42], a convex optimization framework is proposed with diversity incorporated in active learning with an arbitrary classifier.

The most recent and related work with diversity maximization in the sample selection is proposed in [13], where the sample selection is modeled as an optimization problem with the following formulation,

$$\hat{f} = \operatorname{argmin}_f -f^T s + \frac{1}{2} f^T K f, \quad \text{s.t.} \sum_{i=1}^{n} f_i = 1, f_i \geq 0, \tag{1}$$

where f is the updated ranking score, vector s is the sample uncertainty, K is a kernel matrix with $K_{i,j} = \exp(-\frac{\|x_i - x_j\|^2}{\sigma^2})$ which measures the similarity between points x_i and x_j. The first term $-f^T s$ penalizes less if samples with high uncertainty also get high ranking scores. With the kernel matrix K in the second term, $f^T K f$, the algorithm tends to give high ranking

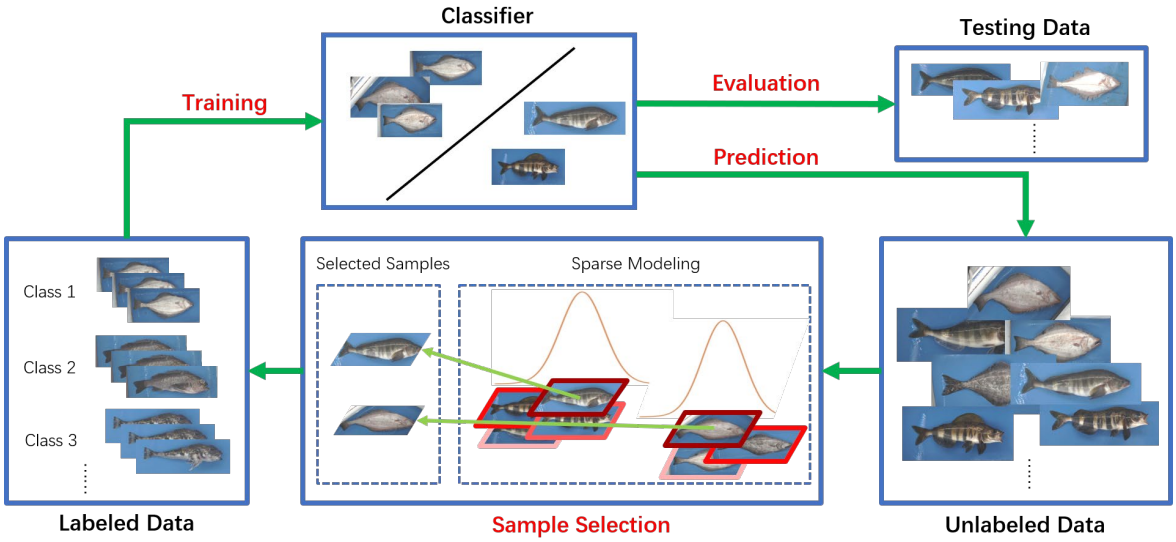


Fig. 1. The flowchart of the learning system.

scores to samples with low similarity. The problem is optimized to find the best trade-off between the uncertainty and the diversity. It shows great power in classification problems with diversity maximization. However, there are two major weaknesses in the algorithm: 1) Isolated distinct samples with high uncertainty are encouraged to be selected. This is because isolated samples are always dissimilar to other samples. Therefore Eq. (1) will generate little penalty on the second term. However, since isolated samples are far away from the data density, these samples are "unimportant" or outliers. Selecting such samples is not very helpful in improving the classifier performance. This strategy results in inefficient selection especially when we are interested in selecting a small batch of samples. 2) The algorithm does not take the batch size into consideration during optimization. In fact, the batch size does matter in the sample selection. Take an extreme situation for example. If the batch size is one, then the sample that lies in the center of the pool data would be the most representative sample. However, if the batch size is two, then we may divide the pool data into two clusters and the sample near the center of each cluster would be the most representative sample. In other words, the selection strategy should vary with the batch size. To address these issues, active learning via sparse modeling is proposed in the following section.

III. Sparse Modeling and an Approximated Solution

The flowchart of our proposed framework is shown in Fig. 1. First, a multi-class SVM classifier is initially trained on the labeled data at the beginning. Then we apply the trained classifier on the unlabeled data. Based on the SVM predictions, sparse modeling via Gaussian kernels is used for sample selection. Then these selected samples are labeled and moved from unlabeled set to labeled set. At the end of each iteration, the classifier is re-trained with the updated labeled set. Finally, the performance of active learning is evaluated on an independent testing dataset. In this section, we will introduce the multi-class SVM classifier, uncertainty measure design, sample selection and an approximated solution to the sparse problem. Note that, in this paper we only use SVM classifiers for our active learning due to the much lower computational complexity requirement, compared to most recent high computational demanding convolution neural networks (CNNs). In fact, the proposed scheme can also be used in many types of classifiers, such as CNNs, if the complexity requirements can be relaxed.

A. Multi-Class SVM Overview

For a multi-class classification problem, we can train linear SVM [32] classifiers based on the "one vs. the rest" strategy. Assume we have K classes. For the k-th class, we treat the training samples that belong to this class as positive samples and all the remaining samples as negative samples. Then the k-th classifier is trained based on the following equation provided in [24],

$$\widehat{w}_k = \arg\min_{w_k \in R^d} C_p \sum_{i=1}^{N} l_2(y_i w_k^T x_i) + \frac{1}{2}\|w_k\|^2, \qquad (2)$$

where $l_2(z)$ is given by $l_2(z) = \max(0,1-z)^2$, $\widehat{w}_k$ are learned weights for the k-th classifier, C_p is a real-valued regularization parameter, and (x_i, y_i) is the i-th instance-label pair. We use l_2 loss instead of hinge-loss to make the training more efficient since the gradient of l_2 loss is continuous. For simplification purposes, we use $\|\cdot\|$ without subscript to denote the l_2 norm $\|\cdot\|_2$. The final SVM classification result can thus be determined by the following equation,

$$\hat{k} = \arg\max_{k \in \{1,2,\ldots,K\}} (\widehat{w}_k^T x_i), \qquad (3)$$

where $\widehat{w}_k^T x_i$ is the prediction of the testing sample x_i corresponding to the k-th class.

B. Uncertainty Measure Design

In active learning, uncertainty sampling aims to choose the most uncertain samples from the unlabeled data pool to label.

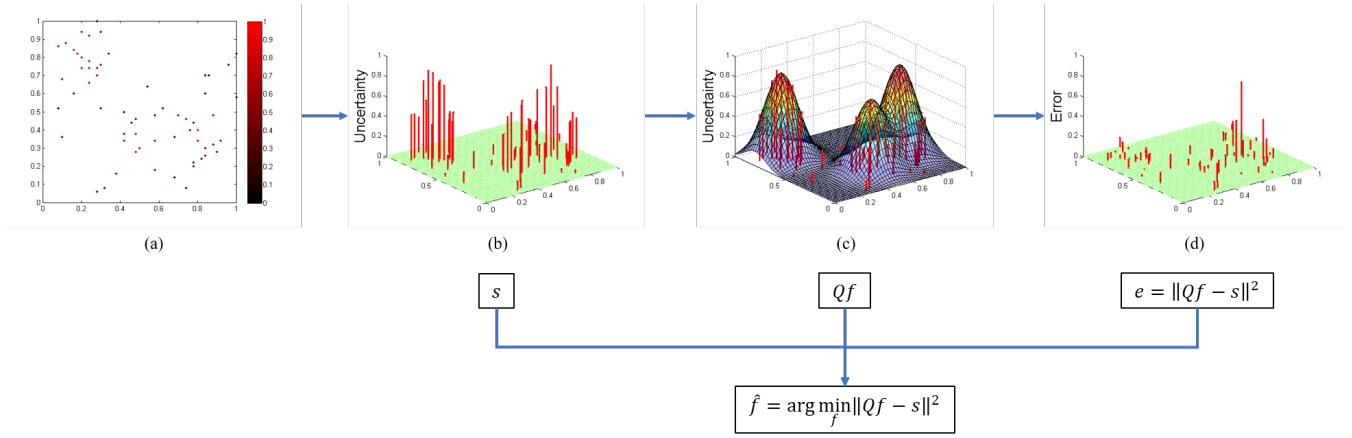


Fig. 2. Overview of the sparse modeling for sample selection. (a): Uncertainty scores for feature points in 2-D space. The color from black to red represents the uncertainty score from low to high. (b): Uncertainty scores are represented in z-axis. (c): Use combination of selected Gaussian kernels to represent the uncertainty scores. f is a sparse vector in which only the indices of selected samples have non-zero values. Q is a collection of Gaussian kernels of all feature points. (d): Representation error with selected Gaussian kernels.

For SVM based classifier, it is common to use the distance between the first two most likely predictions. Similar to [10,11], we define the uncertainty score based on the "best vs. the second best" (BvSB) strategy,

$$s_{\text{BvSB}}(x_i) = \max(\hat{w}_{k_2}^T x_i - \hat{w}_{k_1}^T x_i + 1, 0), \tag{4}$$

where k_1 and k_2 are the first two most likely predicted classes. We take $\max(\cdot)$ operation to restrict the uncertainty score in the range of [0, 1].

C. *Sample Selection via Sparse Modeling*

Given uncertainty scores generated from the classifiers, we would like to select the most informative samples for a query. The simplest selection strategy is that we always select the samples up to the batch size, B_q, with the highest uncertainty scores. However, this strategy ignores the relations among the pooled unlabeled samples. Sometimes the samples with top uncertainty are very similar to each other. We should avoid selecting samples with redundant information in the same batch.

To achieve this goal, we can formulate the problem via sparse representation as shown in Fig. 2. In other words, we want to select a few samples that can cover the information of the pool data as much as possible. To be specific, we propose the following formulation to modify the uncertainty scores before sample selection,

$$\begin{aligned} &\hat{f} = \arg\min_f \|Qf - s\|^2, \\ &s.t. \|f\|_0 = B_q,\ \mathbf{0} \le f \le \mathbf{1}, \end{aligned} \tag{5}$$

where s is the original uncertainty score, $\hat{f}$ is the modified uncertainty score, $\|f\|_0 = \text{card}(f)$ represents the number of non-zeros entries, $\mathbf{0}$ and $\mathbf{1}$ are all-zero vector and all-one vector, respectively, B_q is the batch size and Q is the similarity matrix among all the unlabeled samples. Specifically, $Q_{i,j}$ represents the similarity between samples i and j in the range of [0, 1]. The similarity can be measured in different ways [56,58]. One common method of designing similarity matrix Q is using the Gaussian kernel of two points, i.e.,

$$Q_{i,j} = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right). \tag{6}$$

However, when dealing with high-dimensional data points, which are commonly very sparse, the Euclidean distance might not be a good choice to represent the similarity. To better represent the similarity between two samples, we define the matrix Q as

$$Q_{i,j} = \begin{cases} \exp\left(-\frac{\|\tilde{x}_i - \tilde{x}_j\|^2}{\sigma^2}\right), & \text{if } i \in N_j, \\ 0, & \text{if } i \notin N_j, \end{cases} \tag{7}$$

$$\tilde{x} = [\hat{w}_1, \hat{w}_2, \dots, \hat{w}_K]^T x, \tag{8}$$

where $\tilde{x}_i$ and $\tilde{x}_j$ are transformed data samples of the initial data x_i and x_j, N_j is the neighbor index set of the j-th sample. Here we use the learned weights of SVM classifiers as the data transform.

Assume there are N_U unlabeled samples. To better illustrate the formulation in Eq. (5), we can write matrix Q as $Q = [q_1, q_2, \dots, q_{N_U}]$ and each column vector q_j denotes the similarity weights between the j-th sample and all unlabeled samples via the Gaussian kernel. In this formulation, we are interested in looking for a sparse linear combination of the similarity weight vectors centered at selected samples, i.e., $Q\hat{f}$, to represent the original uncertainty scores s. After the problem is solved, the indices of non-zero entries in $\hat{f}$ would be the indices of our selected samples.

D. *Approximated Solution 1: Greedy Search*

The solution to the problem in Eq. (5) can be well approximated using greedy search method, i.e., we can select samples one-by-one and modify the uncertainty scores after

each selection. Note that this greedy search method still follows the batch mode setting since there is no need to update the classifier after each sequential selection. We denote the similarity matrix Q as $Q = [q_1, q_2, \dots, q_{N_U}]$, where each column vector q_j in Q represents Gaussian kernel weights centered at the location of $\tilde{x}_j$. For the t-th selection from 1 to B_q, the selection strategy is as follows,

$$\hat{k}^t, \hat{f}_{\hat{k}^t} = \arg\min_{j \in U, f_j} \left\| f_j q_j - s^t \right\|^2, \tag{9}$$

where s^t is a vector of uncertainty scores of all unlabeled samples at time t, f_j is a scalar which represents the modified uncertainty score of the j-th sample, U is the index set of unlabeled data, $\hat{k}^t$ is the index of selected sample and $\hat{f}_{\hat{k}^t}$ is the modified uncertainty score for the selected sample.

This can be solved by sequentially obtaining $\hat{k}^t$ and $\hat{f}_{\hat{k}^t}$ using

$$\hat{k}^t = \arg\max_{j \in U} q_j^T s^t, \tag{10}$$

$$\hat{f}_{\hat{k}^t} = \arg\min_{f_{\hat{k}^t}} \left\| f_{\hat{k}^t} q_{\hat{k}^t} - s^t \right\|^2. \tag{11}$$

In Eq. (10), the sample with the maximum correlation between the Gaussian kernel q_j and uncertainty score s is selected. Then the modified uncertainty of the selected sample is calculated from Eq. (11).

After each selection, the remaining uncertainty is calculated from

$$s^{t+1} = \max(s^t - \hat{f}_{\hat{k}^t} q_{\hat{k}^t}, 0). \tag{12}$$

For each iteration, we keep the uncertainty score s^{t+1} to be non-negative. Then we move $\hat{k}^t$ from the unlabeled set U to the labeled set L. This greedy search method is similar to orthogonal matching pursuit (OMP) [33], except that we only keep non-negative values for residuals in Eq. (12). The approach is summarized in Algorithm 1. We name this method as sparse modeling by greedy search (SMGS).

Algorithm 1: SMGS

Input: original uncertainty score s, similarity matrix Q, labeled set L, unlabeled set U.
Initialization: Set $s^1 = s$.
for $t = 1: B_q$ **do**
  Choose $\hat{k}^t = \arg\max_{j \in U} q_j^T s^t$ from U for a query.
  Compute $\hat{f}_{\hat{k}^t}$ by
    $\hat{f}_{\hat{k}^t} = \arg\min_{f_{\hat{k}^t}} \left\| f_{\hat{k}^t} q_{\hat{k}^t} - s^t \right\|^2$.
  Update the uncertainty scores of the next iteration using
    $s^{t+1} = \max(s^t - \hat{f}_{\hat{k}^t} q_{\hat{k}^t}, 0)$.
  Move sample index $\hat{k}^t$ from U to L.
end for
Output: updated labeled set L.

Although the sparse modeling problem can be approximated using Algorithm 1, there are still three major drawbacks of the formulation in Eq. (5): 1) the sparse representation is sensitive to the samples with low uncertainty scores; 2) the uncertainty, diversity and density are not well combined in the formulation; 3) optimal solution is not guaranteed using greedy search method. We will illustrate how we can overcome these drawbacks in the following section.

IV. Combine Uncertainty, Diversity and Density with l_1 Approximation

Sparse modeling is a good way to incorporate diversity and density in the sample selection. However, it is sensitive to the samples with low uncertainty scores. A pre-processing step, i.e., selective sampling, can address this problem before sparse modeling. In addition to diversity and density, we still need to focus on high uncertainty samples. Hence, a trade-off among diversity, density and uncertainty cannot be avoided. Moreover, an efficient approximation is needed for solving the sparse problem with l_0 norm. In this section, selective sampling, modification of the sparse modeling and an efficient optimization approach are proposed.

A. Selective Sampling for Sparse Modeling

For multi-class classification problems, there is often the case that samples with high similarity may have a large difference in the uncertainty. This situation results from the non-robust classifier due to the limited training data. Therefore, the neighboring samples for a given selected sample may have large difference in the uncertainty. Once we apply a Gaussian similarity kernel on a given sample, the samples with high uncertainty cannot be well represented by the kernel if several low uncertainty samples are around. This is because the loss function defined in Eq. (5) is to minimize the representation error of all samples including low uncertainty samples as illustrated in the example given in Fig. 3. From the top-right figure of Fig. 3, we can see that low uncertainty samples can have a large effect on the representation error. With the selective sampling strategy adopted in bottom-left of Fig. 3, the samples with low uncertainty are filtered out before the sparse modeling, resulting in lower representation errors as shown in bottom-right of Fig. 3.

To implement this selective sampling strategy shown in Fig. 3, we design a locality thresholding method to select high uncertainty samples among neighboring samples. Given an unlabeled sample j, we compare it with its neighboring unlabeled samples $i \in N_j$. We define the uncertainty influence, $I_{i,j}$, as a weighted uncertainty score from the sample i to the sample j, i.e., $I_{i,j} = Q_{i,j} s_i$. If $I_{i,j}$ has a much higher value than s_j, we should not select the sample j since it has a much lower

uncertainty than its neighboring samples. Let's define the influence difference d_j as,

$$d_j = \max_{i \in N_j} I_{i,j} - s_j = \max_{i \in N_j} Q_{i,j} s_i - s_j. \quad (13)$$

diversity in the sample selection, where $A = Q^T Q$ is a positive semi-definite matrix with $A_{i,j}$ measuring the similarity between samples i and j. We can see that if $A_{i,j}$ has a high value and both i and j have been selected, then $f_i A_{i,j} f_j$ would have a very high value, which leads to a heavy penalty on the loss function. As a result, this term guarantees that samples with high similarities cannot be selected at the same time, i.e., diverse

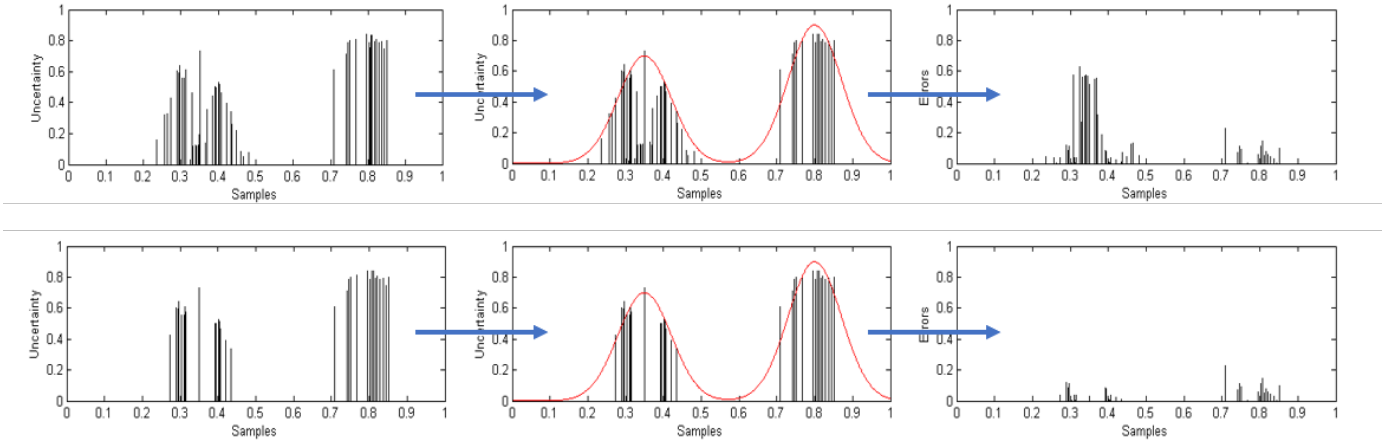


Fig. 3. An example of selective sampling. The first row shows the sparse modeling without selective sampling, which results in high representation error. The second row shows the effectiveness with selective sampling before sparse modeling, which can largely reduce the representation error.

To be specific, the samples with $d_j > d_{\text{thresh}}$ are filtered out, where d_{thresh} is a pre-defined threshold. In other words, we select samples with the uncertainty scores that are not much lower than the uncertainty of the neighboring samples. We denote the index set of selected samples as S_U. Note that this pre-selection step filters out samples in their local neighbors instead of using a global threshold, which is more suitable for sparse representation.

There are two advantages of this selective sampling strategy. On one side, the sparse representation is no longer influenced by low uncertainty samples. On the other side, the number of candidate samples is largely reduced, which also reduces the complexity in the optimization.

B. Combine Diversity, Density and Uncertainty

In this subsection, we will demonstrate how we combine diversity, density and uncertainty by modified sparse modeling. After selective sampling, we only focus on a subset of the unlabeled samples, i.e., $S_U = \{k_1, k_2, \dots, k_m\}$ with $m = \text{card}(S_U)$. Hence, we modify the variables in Eq. (5) with $s = [s_{k_1}, s_{k_2}, \dots, s_{k_m}]^T$ and $Q = [q_{k_1}, q_{k_2}, \dots, q_{k_m}]$. Moreover, we rewrite Eq. (5) as,

$$\begin{aligned} \hat{f} &= \arg\min_f \|Qf - s\|^2, \\ &= \arg\min_f \tfrac{1}{2} f^T Q^T Q f - f^T Q^T s \\ s.t.\ &\|f\|_0 = B_q,\ \mathbf{0} \le f \le \mathbf{1}. \end{aligned} \quad (14)$$

We analyze the above formulation in three aspects as follows.

(1) Diverse term. The first term $\frac{1}{2} f^T Q^T Q f$ measures the diversity in the sample selection, where $A = Q^T Q$ is a positive semi-definite matrix with $A_{i,j}$ measuring the similarity between samples i and j. We can see that if $A_{i,j}$ has a high value and both i and j have been selected, then $f_i A_{i,j} f_j$ would have a very high value, which leads to a heavy penalty on the loss function. As a result, this term guarantees that samples with high similarities cannot be selected at the same time, i.e., diverse samples are encouraged to be selected.

(2) Density term. The second term, $-f^T Q^T s$, measures the density in the sample selection. We can treat $(Qf)^T s$ as a correlation between a combination of selected Gaussian kernels and the uncertainty scores s. If we have a large density of samples around selected samples, then there would be a high correlation between Qf and s. Therefore, there is only a small penalty on the loss function.

(3) Uncertainty trade-off. To emphasize high uncertainty samples, an uncertainty term, $-f^T s$, can be added to strengthen the role of uncertainty in the sample selection as shown in Eq. (15).

We can relax the density term and uncertainty term with penalty parameters λ_1 and λ_2, so that the modified formulation of the sparse modeling becomes,

$$\begin{aligned} \hat{f} &= \arg\min_f \tfrac{1}{2} f^T Q^T Q f - \lambda_1 f^T Q^T s - \lambda_2 f^T s, \\ &= \arg\min_f \tfrac{1}{2} f^T Q^T Q f - f^T (\lambda_1 Q^T + \lambda_2 I) s, \\ s.t.\ &\|f\|_0 = B_q,\ \mathbf{0} \le f \le \mathbf{1}. \end{aligned} \quad (15)$$

To better demonstrate the difference between Eq. (15) and Eq. (1) [13], we make some detailed analyses as follows.

- **Density analysis**. Eq. (1) in [13] does not use the density term of Eq. (15), $-f^T Q^T s$. As a result, isolated distinct samples are encouraged to be selected in Eq. (1). This is because isolated samples are dissimilar to other samples, therefore Eq. (1) will generate a small penalty on the diverse term. However, since isolated samples are far away from the data density, these samples are

"unimportant" or outliers. Selecting such samples is not very helpful for the classifier to improve the performance.

- **Sparsity analysis**. Eq. (15) is derived from sparse representation with only selected samples being considered in the linear combination. As a result, the number of batch size, B_q , is incorporated in our formulation. In other words, the optimal solution is determined with B_q as a hyper-parameter. Different from the proposed method, Eq. (1) is not derived from sparse modeling and therefore the sparsity in their formulation is never analyzed. Besides, they do not incorporate the batch size in their optimization.
- **Efficiency analysis**. In our formulation, we construct matrix Q with only k-nearest neighbors, as illustrated in Eq. (7). As a result, this setting makes the quadratic matrix Q^TQ become a sparse matrix, which leads to efficient optimization [52]. Moreover, the selective sampling step can also largely reduce the searching space. On the contrary, the k-nearest neighbors strategy cannot be easily adopted in matrix K of Eq. (1). This is because this setting will make K become a nonsymmetric matrix. Therefore, the convexity of the formulation will no longer hold, which only results in local minimum solution.

C. Approximated Solution 2: QP via l_1 Norm Relaxation

We are interested in the sparse solution of Eq. (15), which is NP-hard since there is an l_0-norm constraint. If we have $\mathrm{card}(S_U)$ pre-selected unlabeled samples, then we should try $\binom{\mathrm{card}(S_U)}{B_q}$ combinations to select the optimal B_q samples for a query, which is not practical. Therefore, an approximated solution is proposed via l_1-norm relaxation, i.e., we can relax $\|f\|_0$ to $\|f\|_1$. As a result, the problem becomes

$$\begin{aligned}&\hat{f} = \arg\min_f \frac{1}{2} f^T Q^T Q f - f^T(\lambda_1 Q^T + \lambda_2 I)s,\\ &s.t.\ \|f\|_1 = B_q,\ \mathbf{0} \le f \le \mathbf{1}.\end{aligned} \tag{16}$$

It is equivalent to

$$\begin{aligned}&\hat{f} = \arg\min_f \frac{1}{2}\ f^T Q^T Q f - f^T(\lambda_1 Q^T + \lambda_2 I)s,\\ &s.t.\ \mathbf{1}^T f = B_q,\ Cf \le d,\end{aligned} \tag{17}$$

where $C = [-I, I]^T$, $d = [\mathbf{0}^T, \mathbf{1}^T]^T$ and I is the identity matrix. Here, we convert the lower and upper bounds of f to be linear inequality constraints. Moreover, $\mathbf{1}^T f = B_q$ is equivalent to $\|f\|_1 = B_q$ because entries in f are all non-negative values. Hence, the formulation becomes a standard quadratic programming (QP) problem, which can now be solved by the interior-point method [39,40]. First, we form the Lagrangian function for the above problem, i.e.,

$$\begin{aligned}L(f, y, z) = &\frac{1}{2} f^T Q^T Q f - f^T(\lambda_1 Q^T + \lambda_2 I)s\\ &- y\left(\mathbf{1}^T f - B_q\right) - z^T(Cf - d),\end{aligned} \tag{18}$$

where y and z are the vectors of Lagrange multipliers. Then the Karush-Kuhn-Tucker (KKT) conditions [55] can be stated as follows,

$$\begin{aligned}&Q^T Q f - (\lambda_1 Q^T + \lambda_2 I)s - y\mathbf{1} - C^T z = 0,\\ &Cf - d + \tau = 0,\\ &\mathbf{1}^T f - B_q = 0,\\ &z_i \tau_i = 0, i = 1,2,\ldots,m,\\ &\tau \ge \mathbf{0},\\ &z \ge \mathbf{0},\end{aligned} \tag{19}$$

where τ is the slack vector that converts inequality constraints to equalities. Then we define the residuals as follows,

$$\begin{aligned}&r_d = Q^T Q f - (\lambda_1 Q^T + \lambda_2 I)s - y\mathbf{1} - C^T z,\\ &r_{eq} = \mathbf{1}^T f - B_q,\\ &r_{ineq} = Cf - d + \tau,\\ &r_{\tau z} = \Gamma z,\end{aligned} \tag{20}$$

where Γ is the diagonal matrix of τ. In a Newton step, the changes in x, τ, y, and z, are given by,

$$\begin{pmatrix} Q^TQ & \mathbf{0} & -\mathbf{1} & -C^T \\ \mathbf{1}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ C & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & Z & \mathbf{0} & \Gamma \end{pmatrix} \begin{pmatrix} \Delta f \\ \Delta\tau \\ \Delta y \\ \Delta z \end{pmatrix} = - \begin{pmatrix} r_d \\ r_{eq} \\ r_{ineq} \\ r_{\tau z} \end{pmatrix}, \tag{21}$$

where Z is the diagonal matrix of z. We update $f_{t+1} = f_t + \Delta f$ for each iteration until convergence. Additionally, if a full Newton step is infeasible, we shorten the step to maintain positivity.

However, the number of non-zero entries in f is not constrained with l_1-norm. Since we restrict f to [0, 1], the solution will give us more than B_q non-zero entries. Besides that, since we are only interested in the first B_q samples, we do not want non-selected samples to influence the solution. As a result, we modify the formulation by introducing a parameter λ in the constraint, i.e.,

$$\begin{aligned}&\hat{f} = \arg\min_f \tfrac{1}{2}\ f^T Q^T Q f - f^T(\lambda_1 Q^T + \lambda_2 I)s,\\ &s.t.,\ \mathbf{1}^T f = \lambda B_q,\ Cf \le d,\end{aligned} \tag{22}$$

where we choose λ between 0 and 1. Decreasing λ will generate less non-zero entries in the solution. We can adjust λ so that the solution only contains B_q non-zero entries. This problem can be solved by using a simple bisection algorithm to iteratively search the proper λ, which is illustrated in Algorithm 2. We name this method as sparse modeling via quadratic

programming (SMQP), which uses quadratic programming to solve the sparse modeling problem. Usually, the optimal solution will be achieved in no more than 10 iterations from our simulations.

Algorithm 2: SMQP

Input: original uncertainty score s, similarity matrix Q, labeled set L, pre-selected unlabeled set S_U.
Initialization: Set $\lambda = 0.5$, $lb = 0$, $ub = 1$.
Solve $\hat{f}$ by Eq. (22).
while $\|\hat{f}\|_0 \neq B_q$ **do**
  if $\|\hat{f}\|_0 > B_q$ **do**
    $ub = \lambda$.
  else
    $lb = \lambda$.
  end if
  $\lambda = (lb + ub)/2$.
  Solve $\hat{f}$ by Eq. (22).
end while
Sort $\hat{f}$ in descending order and move the first B_q indices of $\hat{f}$ from S_U to L.
Output: updated labeled set L.

V. Experimental Results and Analysis

A. Experiment Setup

We evaluate our proposed method for classification problems on four image datasets, which are COIL-20 [25], a subset of MNIST [27], Cam-Trawl Fish [28] and Chute Fish [29]. The information of the datasets is described in TABLE I.

TABLE I
DATASET DESCRIPTION

Name	# of samples	# of class
COIL-20	1440	20
MNIST (subset)	3000	10
Cam-Trawl Fish	1026	5
Chute Fish	5032	27

For each dataset, we split the data into 4 parts: seed set (labeled set), unlabeled set, validation set and testing set, denoted as L, U, V, T, respectively. L is used for training the initial classifiers; U is treated as a data pool for sample selection; V is used for parameter tuning; and T is used for evaluating the performance of the re-trained classifiers. For each dataset, we split the data as follows: in each class, 3 samples are used as the seed set; half of the samples are used as unlabeled data; one quarter of samples are used as the validation set and the remaining samples are used as the testing set. During each experiment, the data is split randomly.

For each dataset, two different types of feature extraction methods are adopted. One is based on the traditional extraction method and the other is convolutional neural networks (CNNs) based [60]. For the traditional extraction method, we resize and concatenate each sample image into one feature vector in dataset COIL-20 and MNIST, named as "concat" in TABLE II; while for Cam-Trawl Fish and Chute Fish datasets, we follow the bag-of-features (BoF) framework [26,31] based on two level codebook learning [30]. For the CNN based feature extraction method, we use the output of pre-logits layer of inception-resnet-v2 [51] as the feature vector in dataset COIL-20, Cam-Trawl Fish and Chute Fish; while for MNIST dataset, we adopt the architecture of two convolutional layers followed by two fully connected layers in Tensorflow official site [54]. Before extracting CNN features, we use transfer learning on each dataset to achieve better feature representation. The feature space used for the four datasets is summarized in TABLE II.

Eight evaluation methods are used in the experiments: 1) BvSB [11], which chooses the B_q samples based on top highest uncertainty scores; 2) RAND, which randomly selects B_q samples for a query; 3) SMGS, which is the proposed approximated approach using sparse modeling via greedy search; 4) VS [12], which incorporates diversity for a query via version space reduction; 5) USDM [13], which is uncertainty sampling based active learning with diversity maximization; 6) SMQP, which is the proposed approximated approach using sparse modeling via quadratic programming; 7) MMC [49], which is active learning with maximum model change; 8) EER [50], which is active learning with expected error reduction.

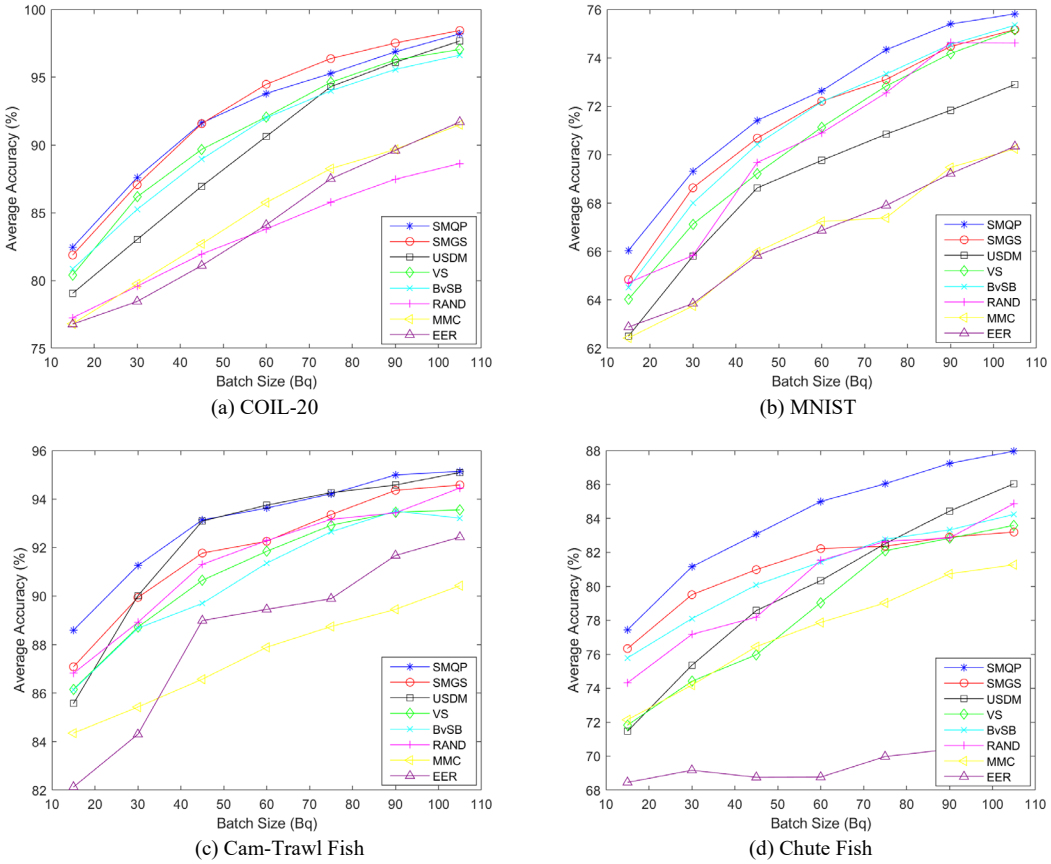
For the parameter settings, we fix the cost $C_p = 1$ in the SVM for all experiments. Also, we fix $d_{\text{thresh}} = 0.2$ in the selective sampling step for the sparse modeling. We also set $\lambda_1 = 1$ in the refinement of sparse modeling fixed. Other parameters are empirically tuned according to the performance in the validation set V. We tune the standard deviation σ of the similarity matrix Q from {0.25, 0.5, 1, 2}, λ_2 in the refinement of sparse modeling from {0.1,1,5,10} and the number of neighbors $\text{card}(N_j)$ of Q from {5,10,20}.

TABLE II
FEATURE DESCRIPTION

Name	Feature size
COIL-20 (concat)	1024
COIL-20 (CNN)	1536
MNIST (concat)	784
MNIST (CNN)	1024
Cam-Trawl Fish (BoF)	7168
Cam-Trawl Fish (CNN)	1536
Chute Fish (BoF)	7168
Chute Fish (CNN)	1536

B. Performance Comparison with Different Batch Sizes

We run the algorithms with different batch sizes from 15 to 105 with 15 increments for each experiment and report the average accuracy for all the eight methods. We set the seed size $c = 3$ in this experiment. Each result is based on an average of 10 runs of the same setting. Figure 4 compares the performance of eight active learning algorithms for image classification on four datasets. Generally, SMQP and SMGS outperform other methods. To be specific, SMQP gives robust results on different batch size and SMGS also gives promising results. However, since no optimal solution is guaranteed in SMGS, in a few cases



(a) COIL-20 (b) MNIST

(c) Cam-Trawl Fish (d) Chute Fish

Fig. 4. Average accuracy with seed size $c = 3$ on four datasets using traditional features.

the accuracy of SMGS is slightly lower than BvSB method, particularly for Chute Fish dataset when the batch size is over 75. For MMC and EER methods, although they are optimized for sequential sample selection, they are not taking advantage of the unlabeled pool data for batch mode settings. We can see in MNIST dataset, they perform even worse than RAND. This is because the sequentially selected samples in MMC and EER have much redundant information which gives little information for re-training the classifiers than random selection. As for VS method, it also incorporates diversity in the experiment design and use version space reduction to deal with binary problem. However, the results show that there is no big improvement than BvSB method. A possible reason is that version space reduction may be not suitable for multi-class problems. Generally, the performance of USDM is usually much better with large batch size than with small batch size. This is because USDM always tends to look for isolated distinct samples at the first few selections, which has been discussed in the previous sections.

C. *Performance Comparison Using Different Features*

In this subsection, we compare the performance of eight active learning methods related to CNN based features. Figure 5 shows the experimental results on the four datasets. More details about CNN feature representation can be found in TABLE II. Since transfer learning is conducted on the dataset before feature extraction, the classifiers are more robust with the same size of seed set compared to using traditional features in the previous subsection. This experiment demonstrates that when a better feature is used, the performance of an active learning algorithm usually improves. Same as before, SMQP outperforms the other competitors consistently using different features. When the batch size increases, the performances saturate among several different methods with robust classifiers.

TABLE III
CONTINGENCY TABLE FOR TWO CLASSIFIERS

		SMQP classifier	
		Correct	Wrong
USDM classifier	Correct	a_1	a_2
	Wrong	a_3	a_4

D. *Performance Comparison Using Different Seed Sizes*

In this subsection, we examine the impact of the seed size by changing $c = 9$ on these four datasets. We keep other settings unchanged and evaluate the performance with the batch sizes varying from 15 to 105. Both traditional features and CNN features are used for complete comparison. The results are shown in Fig. 6 and Fig. 7. From the results, we can see that the proposed methods are consistently favorable, which further indicates that leveraging the unlabeled pool data does help improve the active learning performance.

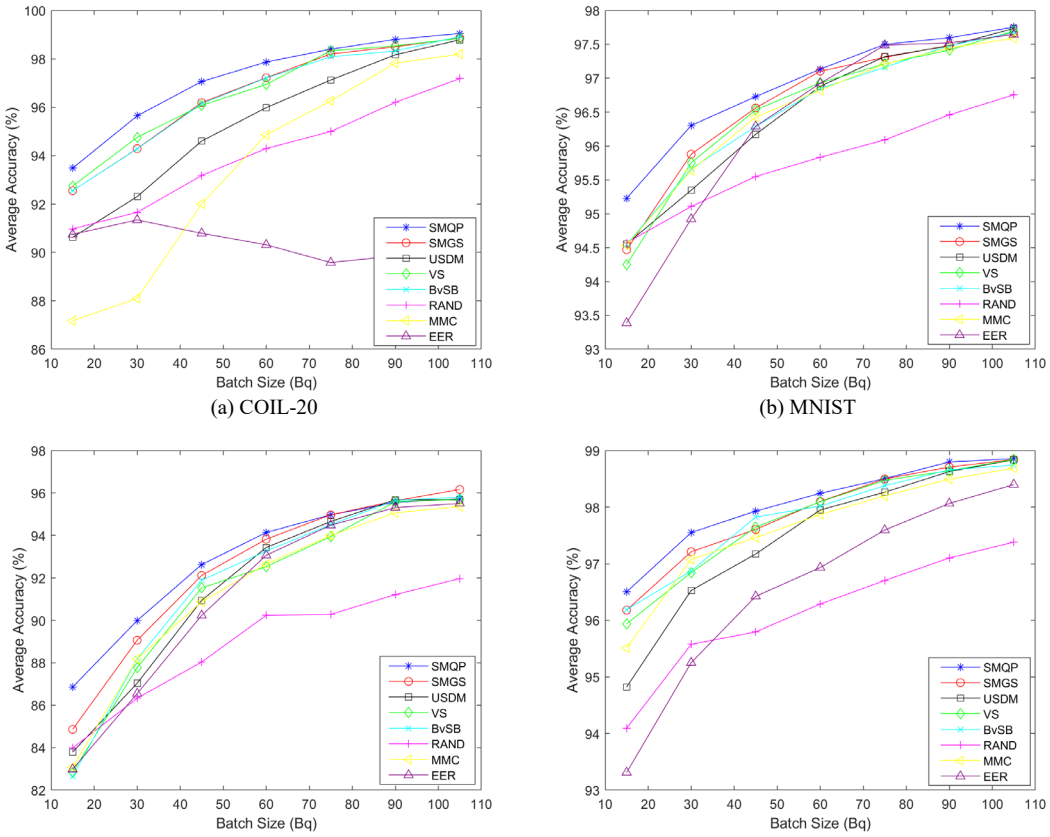
E. *Significance Test Analysis*

Since USDM also incorporates diversity in the minimization, we use paired sign test to verify whether SMQP has a significant improvement over USDM. For each testing sample,

the outcome of the two classifiers have 4 possibilities: 1) both USDM and SMQP make correct predictions; 2) USDM makes a correct prediction while SMQP makes a wrong prediction; 3) SMQP makes a correct prediction while USDM makes a wrong prediction; 4) both USDM and SMQP make wrong predictions. We use a 2×2 contingency table to tabulate the outcomes of two classifiers on all testing samples, as follows.

The null hypothesis H_0 is that these two classifiers are the same while the alternate hypothesis H_1 is that SMQP is better than USDM. The p-value is defined as the probability that the same as or more extreme cases than the actual observed results occur, when the null hypothesis is true. A smaller p-value means that SMQP classifier is more likely to be better than USDM classifier. The p-value can be formulated as follows,

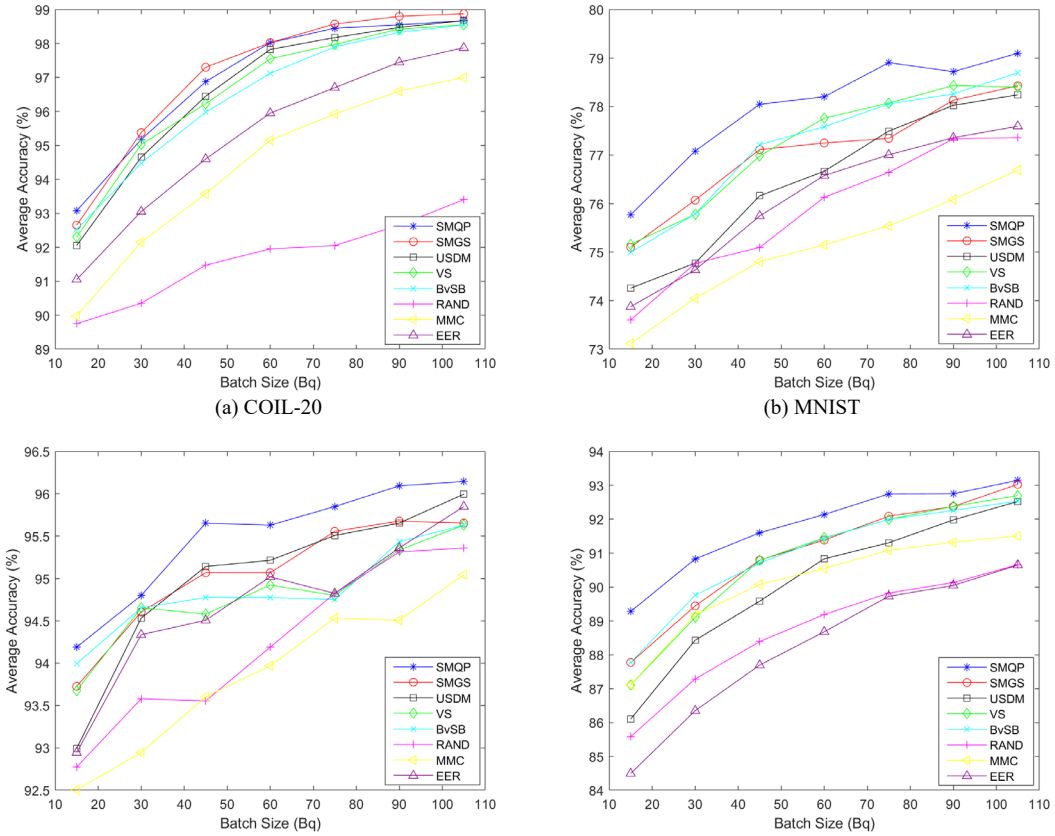
$$\begin{aligned} p &= \Pr(X \geq a_3) = 1 - \Pr(X < a_3) \\ &= 1 - \sum_{i=0}^{a_3-1} \binom{a_2+a_3}{i} 0.5^i \, (1-0.5)^{a_2+a_3-i}. \end{aligned} \tag{23}$$



(a) COIL-20 (b) MNIST

(c) Cam-Trawl Fish (d) Chute Fish

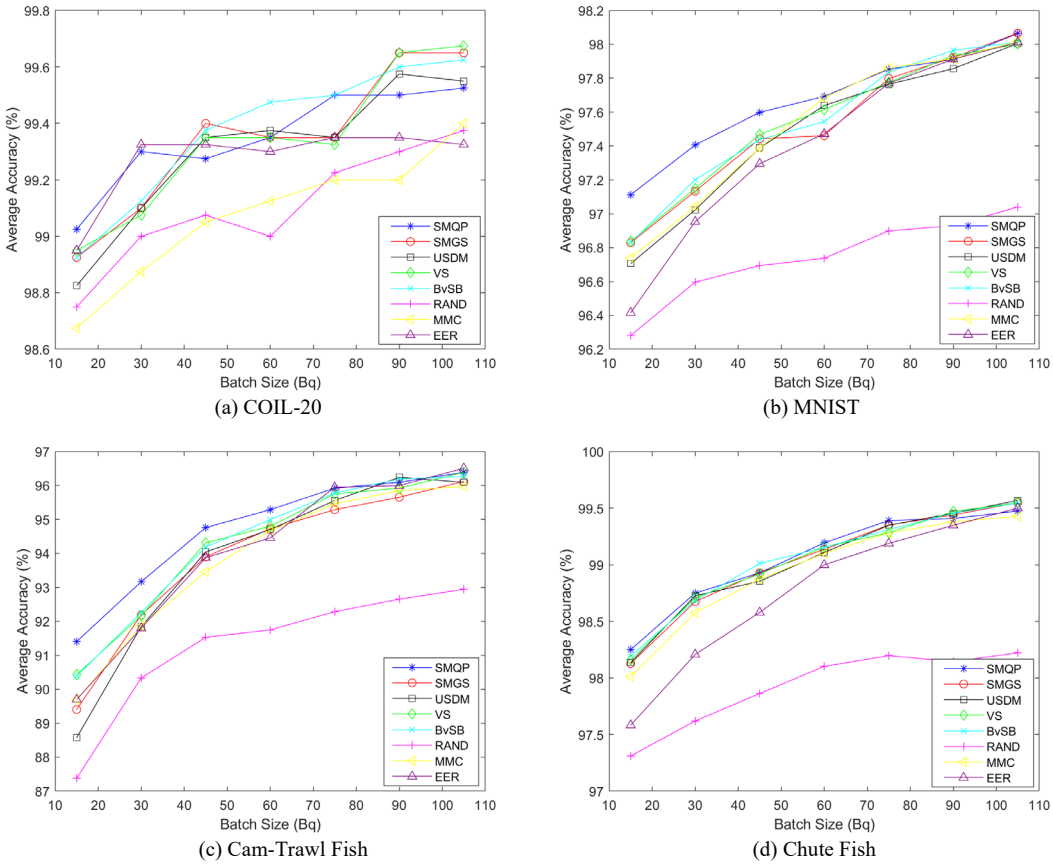
Fig. 5. Average accuracy with seed size $c = 3$ on four datasets using CNN features.



(a) COIL-20 (b) MNIST

(c) Cam-Trawl Fish (d) Chute Fish

Fig. 6. Average accuracy with seed size $c = 9$ on four datasets using traditional features.



(a) COIL-20 (b) MNIST

(c) Cam-Trawl Fish (d) Chute Fish

Fig. 7. Average accuracy with seed size $c = 9$ on four datasets using CNN features.

P-VALUES BETWEEN SMQP AND USDM FOR FOUR DATASETS WITH DIFFERENT BATCH SIZES

Bq	15	30	45	60	75	90	105
COIL-20 (c)	3.37E-11	4.55E-15	0	9.83E-12	4.64E-03	1.90E-02	2.87E-02
COIL-20 (3×c)	4.10E-04	2.31E-02	3.36E-02	8.13E-02	7.75E-02	6.40E-02	5.00E-01
MNIST (c)	7.23E-10	0	0	0	0	0	0
MNIST (3×c)	8.44E-09	3.44E-15	2.38E-12	7.77E-10	3.96E-08	1.18E-03	7.42E-02
Cam-Trawl Fish (c)	7.69E-05	2.69E-04	4.55E-02	8.23E-02	8.68E-02	9.75E-02	5.00E-02
Cam-Trawl Fish (3×c)	8.60E-03	1.10E-03	5.42E-03	5.39E-02	2.02E-02	5.63E-02	3.85E-02
Chute Fish (c)	0	0	0	0	0	0	1.11E-16
Chute Fish (3×c)	0	0	0	0	0	1.87E-08	1.51E-07

We report all p-values based on the sign test between USDM and SMQP classifiers on four datasets using traditional features in Table IV.

For SMQP method, the smaller the p-value is, the more significant the improvement is over USDM method. From the table, we can see that SMQP method has a significant improvement over USDM method especially for small batch size and seed number. With large batch size and seed number, some p-values are relatively large. This is because the classifier becomes more robust with the increase of the batch size and seed number.

F. Performance Comparison of Data Transform in Similarity Matrix

In this subsection, we examine the impact of the data transform in the construction of similarity matrix in Eq. (8). Taking BoF features on Cam-Trawl Fish dataset for example, we report the average accuracy with seed size $c = 3$ in Fig. 8. The dashed line represents the result without data transform, i.e., we use the original features with Euclidean distance to construct the similarity in Eq. (7). On the contrary, the solid line shows the result with the data transform using Eq. (8). As expected, the result with data transform outperforms the one without data transform. This is because the transformed data can better describe the similarity relations among the samples.

Recently, the negativity of data is analyzed [63] and simultaneously updated with the classifier since some negative samples may look more like positive samples than others. As a result, the negativity of samples is not equally weighted, which can be used as a good way for measuring the similarity among samples. This could be one direction about our future work.

G. Effectiveness with Selective Sampling

We take MNIST dataset as an example to show the effectiveness with selective sampling. As shown in blue dashed

curve in Fig. 9, the average accuracy drops slightly without selective sampling. This is because the low uncertainty samples may have negative impact on the sparse representation. The most important effect of selective sampling is the ability of reducing the computation time, which is shown in Fig. 10.

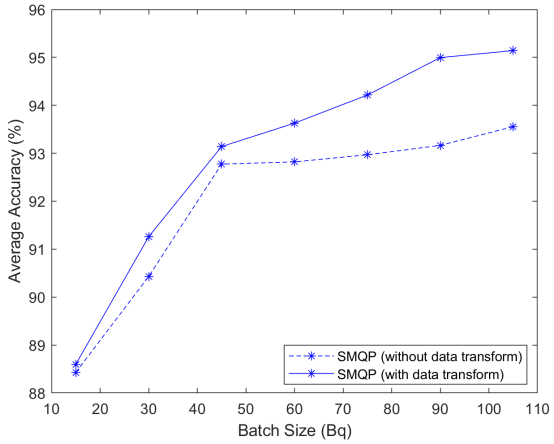


Fig. 8. The performance comparison with and without data transform.

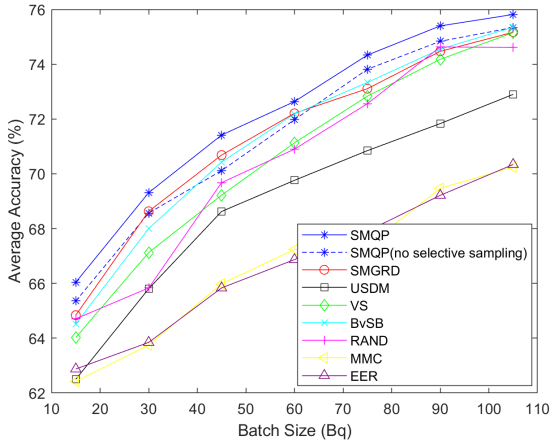


Fig. 9. The effectiveness of selective sampling in MNIST dataset using concatenated features with $c = 3$.

H. Computational Efficiency Comparison

Taking MNIST dataset as an example, we compare the computational efficiency of the sample selection process of nine methods which also includes SMQP without selective sampling. In this experiment, we vary the unlabeled pool size from 300 to 1500, with an interval of 300. All experiments are implemented by Matlab R2017b, which is installed on a machine with 4 core i7 and 32.0GB RAM.

Figure 10 shows the elapsed time to select 15 data for labeling. We can see that except SMQP and USDM, the elapse time remains flat with the increase of the pool size. The dashed blue curve shows the elapsed time using SMQP without selective sampling. Compared with USDM, SMQP without selective sampling is computationally expensive since it also searches the optimal λ in the optimization. As for SMQP, it outperforms USDM in efficiency with the increase of pool size. This is because SMQP adopts selective sampling strategy that drops low uncertainty samples before the optimization.

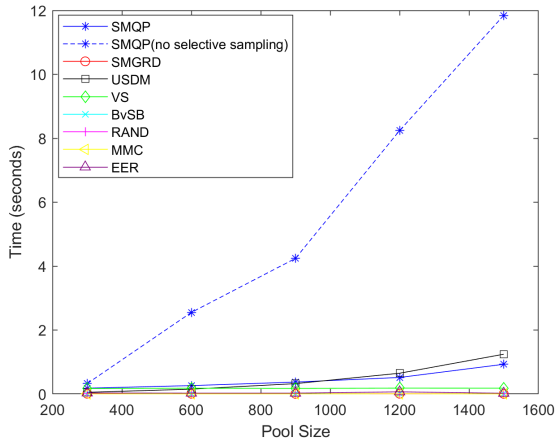


Fig. 10. The elapsed time comparison.

VI. Conclusion and Future Work

In this paper, a novel uncertainty sampling based active learning algorithm is proposed via sparse modeling. An approximated solution by greedy search method is achieved. Moreover, uncertainty, diversity and density are combined in the joint optimization after refinement of the sparse modeling. To overcome the ineffectiveness of solving l_0-norm constraint of the sparse problem, a relaxation of l_1-norm solution, SMQP, is provided by quadratic programming. Comprehensive experiments are conducted with regard to batch size, feature space, seed size, significant analysis, data transform and computational efficiency. There are two directions for future work. On one side, we will look for more effective ways to measure the similarity among samples, such as generating fine-grained labels inspired from [63]. On the other side, we will focus on the sample selection methods when facing large-scale datasets.

References

[1] Q. Zhao and D. J. Miller, "Mixture modeling with pairwise, instancelevelclass constraints," Neural Comput., vol. 17, no. 11, pp. 2482–2507, 2005.

[2] M. Guillaumin, J. Verbeek, and C. Schmid, "Multimodal semisupervised learning for image classification," in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010, pp. 902–909.

[3] Y. Luo, D. Tao, B. Geng, C. Xu, and S. J. Maybank, "Manifold regularized multitask learning for semi-supervised multilabel image classification," IEEE Trans. Image Process., vol. 22, no. 2, pp. 523–536, 2013.

[4] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Large scale transductive SVMs," J. Mach. Learn. Res., vol. 7, no. Aug, pp. 1687–1712, 2006.

[5] J.-N. Hwang, J. J. Choi, S. Oh, and R. J. Marks, "Query-based learning applied to partially trained multilayer perceptrons," IEEE Trans. Neural Netw., vol. 2, no. 1, pp. 131–136, 1991.

[6] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," Mach. Learn., vol. 15, no. 2, pp. 201–221, 1994.

[7] C. Campbell, N. Cristianini, A. Smola, and others, "Query learning with large margin classifiers," in ICML, 2000, pp. 111–118.

[8] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in ICML, 2000, pp. 839–846.

[9] Y. Leng, X. Xu, and G. Qi, "Combining active learning and semisupervised learning to construct SVM classifier," Knowl.-Based Syst., vol. 44, pp. 121–131, 2013.

[10] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 2009, pp. 2372–2379.

[11] A. J. Joshi, F. Porikli, and N. P. Papanikolopoulos, "Scalable active learning for multiclass image classification," IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 11, pp. 2259–2273, 2012.

[12] K. Brinker, "Incorporating diversity in active learning with support vector machines," in ICML, 2003, vol. 3, pp. 59–66.

[13] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, "Multi-class active learning by uncertainty sampling with diversity maximization," Int. J. Comput. Vis., vol. 113, no. 2, pp. 113–127, 2015.

[14] L. Zhang, C. Chen, J. Bu, D. Cai, X. He, and Thomas S. Huang, "Active learning based on locally linear reconstruction," IEEE Transactions on Pattern Analysis and Machine Intelligence 33, no. 10 (2011): 2026-2038.

[15] H. Zhang, H. BvSB, M. Kong, H. Fang, and Z. Zhao, "Active Learning with Sparse Reconstruction."

[16] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," Journal of artificial intelligence research (1996).

[17] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study," Knowledge and Information Systems 42, no. 2 (2015): 245-284.

[18] N. Fazakis, S. Karlos, S. Kotsiantis, and K. Sgarbas, "Self-trained LMT for semisupervised learning," Computational intelligence and neuroscience 2016 (2016): 10.

[19] Y. Hu, D. Zhang, Z. Jin, D. Cai, and X. He, "Active learning via neighborhood reconstruction," In Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, pp. 1415-1421. AAAI Press, 2013.

[20] K. Yu, J. Bi, and V. Tresp, "Active learning via transductive experimental design," In Proceedings of the 23rd international conference on Machine learning, pp. 1081-1088. ACM, 2006.

[21] H. Schütze, E. Velipasaoglu, and J. O. Pedersen, "Performance thresholding in practical text classification," In Proceedings of the 15th ACM international conference on Information and knowledge management, pp. 662-671. ACM, 2006.

[22] K. Tomanek, and U. Hahn, "A comparison of models for cost-sensitive active learning," In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 1247-1255. Association for Computational Linguistics, 2010.

[23] B. C. Wallace, K. Small, C. E. Brodley, and T. A. Trikalinos, "Active learning for biomedical citation screening," In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 173-182. ACM, 2010.

[24] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," J. Mach. Learn. Res., vol. 9, no. Aug, pp. 1871–1874, 2008.

[25] S. A. Nene, S. K. Nayar, H. Murase, and others, "Columbia object image library (COIL-20)," 1996.

[26] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in Workshop on statistical learning in computer vision, ECCV, 2004, vol. 1, pp. 1–2.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[28] M.-C. Chuang, J.-N. Hwang, K. Williams, and R. Towler, "Automatic fish segmentation via double local thresholding for trawl-based underwater camera systems," In Image Processing (ICIP), 2011 18th IEEE International Conference on, pp. 3145-3148. IEEE, 2011.

[29] T.-W. Huang, J.-N. Hwang, and C. S. Rose, "Chute based automated fish length measurement and water drop detection," In Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, pp. 1906-1910. IEEE, 2016.

[30] G. Wang, J.-N. Hwang, K. Williams, F. Wallace, and C. S. Rose, "Shrinking Encoding with Two-Level Codebook Learning for Fine-Grained Fish Recognition," In Computer Vision for Analysis of Underwater Imagery (CVAUI), 2016 ICPR 2nd Workshop on, pp. 31-36. IEEE, 2016.

[31] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," In Computer vision and pattern recognition, 2006 IEEE computer society conference on, vol. 2, pp. 2169-2178. IEEE, 2006.

[32] C. Cortes, and V. Vapnik, "Support-vector networks," Machine learning 20, no. 3: 273-297, 1995.

[33] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," In Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on, pp. 40-44. IEEE, 1993.

[34] V. Sindhwani, P. Niyogi, M. Belkin, and S. Keerthi, "Linear manifold regularization for large scale semi-supervised learning," Proc. of the 22nd ICML Workshop on Learning with Partially Classified Training Data. Vol. 28. 2005.

[35] S. C. H. Hoi, R.Jin, J. Zhu, and M. R. Lyu, "Semisupervised SVM batch mode active learning with applications to image retrieval," ACM Transactions on Information Systems (TOIS) 27.3: 16, 2009.

[36] Z. Xu, R. Akella, and Y. Zhang, "Incorporating diversity and density in active learning for relevance feedback," ECiR. Vol. 7. 2007.

[37] A. Gadde, A. Anis, and A. Ortega, "Active semi-supervised learning using sampling theory for graph signals," Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2014.

[38] E. Pasolli, F. Melgani, D. Tuia, F. Pacifici, and W. J. Emery, "SVM active learning approach for image classification using spatial information," IEEE Transactions on Geoscience and Remote Sensing 52.4: 2217-2233, 2014.

[39] A. Altman, and J. Gondzio, "Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization," Optimization Methods and Software11.1-4: 275-302, 1999.

[40] R. J. Vanderbei, and T. J. Carpenter, "Symmetric indefinite systems for interior point methods," Mathematical Programming 58.1-3: 1-32, 1993.

[41] B. Settles, "Active learning," Synthesis Lectures on Artificial Intelligence and Machine Learning 6.1: 1-114, 2012.

[42] E. Elhamifar, G. Sapiro, A. Yang, and S. S. Sasrty, "A convex optimization framework for active learning," In Computer Vision (ICCV), 2013 IEEE International Conference on pp. 209-216, 2013.

[43] L. Lin, K. Wang, D. Meng, W. Zuo, and L. Zhang, "Active self-paced learning for cost-effective and progressive face identification," IEEE transactions on pattern analysis and machine intelligence, 40(1), 7-19, 2018.

[44] L. Liang, and K. Grauman, "Beyond comparing image pairs: Setwise active learning for relative attributes," In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 208-215), 2014.

[45] K. Wang, Zhang, D., Li, Y., Zhang, R., & Lin, L, "Cost-effective active learning for deep image classification," IEEE Transactions on Circuits and Systems for Video Technology, 2016.

[46] R. Tudor Ionescu, B. Alexe, M. Leordeanu, M. Popescu, D. P. Papadopoulos, and V. Ferrari, "How hard can it be? Estimating the difficulty of visual search in an image," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2157-2166), 2016.

[47] Y. Yan, F. Nie, W. Li, C. Gao, Y. Yang, and D. Xu, "Image classification by cross-media active learning with privileged information," IEEE Transactions on Multimedia, 18(12), 2494-2502, 2016.

[48] S. Vijayanarasimhan, and K. Grauman, "Large-scale live active learning: Training object detectors with crawled data and crowds," International Journal of Computer Vision, 108(1-2), 97-114, 2014.

[49] W. Cai, Y. Zhang, S. Zhou, W. Wang, C. Ding, and X. Gu, "Active learning for support vector machines with maximum model change," In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 211-226). Springer, Berlin, Heidelberg, 2014.

[50] B. Yang, J. T. Sun, T. Wang, and Z. Chen, "Effective multi-label active learning for text classification," In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 917-926). ACM, 2009.

[51] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," In AAAI (Vol. 4, p. 12), 2017.

[52] C. C. Paige, and M. A. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," ACM transactions on mathematical software, 8(1), 43-71, 1982.

[53] G. Wang, J. N. Hwang, C. Rose, and F. Wallace, "Uncertainty sampling based active learning with diversity constraint by sparse selection," In Multimedia Signal Processing (MMSP), IEEE 19th International Workshop on (pp. 1-6). IEEE, 2017.

[54] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, ... and S. Ghemawat, "TensorFlow: large-scale machine learning on heterogeneous systems," Software available from tensorflow. Org. URL https://www.tensorflow.org/versions/r1.2/get_started/mnist/pros.

[55] H. W. Kuhn, and A. W. Tucker, "Nonlinear programming," In Traces and emergence of nonlinear programming (pp. 247-258). Birkhäuser, Basel, 2014.

[56] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(10), 2007.

[57] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," IEEE transactions on pattern analysis and machine intelligence, 28(7), 1088-1099, 2006.

[58] D. Tao, X. Li, X. Wu, and S. J. Maybank, "Geometric mean for subspace selection," IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(2), 260-274, 2009.

[59] B. Du, Z. Wang, L. Zhang, L. Zhang, and D. Tao, "Robust and discriminative labeling for multi-label active learning based on maximum correntropy criterion," IEEE Transactions on Image Processing, 26(4), 1694-1707, 2017.

[60] B. Du, W. Xiong, J. Wu, L. Zhang, L. Zhang, and D. Tao, "Stacked convolutional denoising auto-encoders for feature representation," IEEE transactions on cybernetics, 47(4), 1017-1027, 2017.

[61] B. Du, Z. Wang, L. Zhang, L. Zhang, W. Liu, J. Shen, and D. Tao, "Exploring representativeness and informativeness for active learning," IEEE transactions on cybernetics, 47(1), 14-26, 2017.

[62] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," Mach. Learn., vol. 15, no. 2, pp. 201–221, 1994.

[63] Z. Ma, X. Chang, Y. Yang, N. Sebe, and A. G. Hauptmann, "The many shades of negativity," IEEE Transactions on Multimedia, 19(7), 1558-1568, 2017.