# Self-Paced Active Learning: Query the Right Thing at the Right Time

**Ying-Peng Tang, Sheng-Jun Huang***
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
Collaborative Innovation Center of Novel Software Technology and Industrialization
Nanjing 211106, China
{tangyp, huangsj}@nuaa.edu.cn

**Abstract**

Active learning queries labels from the oracle for the most valuable instances to reduce the labeling cost. In many active learning studies, informative and representative instances are preferred because they are expected to have higher potential value for improving the model. Recently, the results in self-paced learning show that training the model with easy examples first and then gradually with harder examples can improve the performance. While informative and representative instances could be easy or hard, querying valuable but hard examples at early stage may lead to waste of labeling cost. In this paper, we propose a self-paced active learning approach to simultaneously consider the potential value and easiness of an instance, and try to train the model with least cost by querying the right thing at the right time. Experimental results show that the proposed approach is superior to state-of-the-art batch mode active learning methods.

## Introduction

In many real world applications, the amount of unlabeled data is far larger than that of labeled data; and label acquisition is expensive and difficult. It is thus rather important to train an effective model with fewer labeled examples. Active learning is one of the main approaches to deal with this challenge. It expects to reduce the labeling cost by selecting the most valuable instances to query their labels from the oracle (Settles 2009). During the past decades, many criteria have been proposed for active selection of instances. Among which, informativeness and representativeness are most frequently used, and their integration has been validated to be effective for selecting the most valuable instances (Wang and Ye 2013; Huang and Zhou 2013; Huang, Jin, and Zhou 2014).

All these criteria try to estimate the potential value of an instance on improving the model performance. However, they neglect the fact that the potential ability of a valuable instance may not be fully exploited at a specific stage of the model training. This phenomenon has been well validated in self-paced learning (SPL) (Kumar, Packer, and Koller 2010).

*This work was supported by National Key R&D Program of China (2018YFB1004300), NSFC (61503182, 61876081, 61732006).


Self-paced learning is inspired by the "easy to hard" process of human learning. It learns the easier instances first, and then gradually adds more complex instances for model into training. A typical SPL model tries to minimize a weighted sum of losses for all instances, where the weight reflects the easiness of an instance to the model. By optimizing the weights and the model alternately, instances are gradually involved from easy to hard. It has been well validated by many studies that this learning paradigm can lead to better generalization performances (Khan, Mutlu, and Zhu 2011; Tang, Yang, and Gao 2012; Basu and Christensen 2013; Zhang et al. 2016).

Based on these observations, it can be implied that at a specific training stage, over-complex examples may be less useful than easy ones for improving the model. On the other hand, while existing active learning methods focus on selecting informative and representative instances, they fail to query the right thing at the right time. Although the selected instances have high potential value for model improving, they may not be fully exploited at the current stage, and thus lead to waste of labeling cost. As a result, in addition to criteria estimating the potential value of improving the model, easiness of an instance to the current model should be considered in active selection.

In this paper, we propose a novel approach called SPAL (Self-Paced Active Learning) to simultaneously consider potential value and easiness of instances. On one hand, the selected instances should be informative and representative; on the other hand, they should not be over-complex for the current model, and thus can be fully utilized. Specifically, we maintain two weights for each unlabeled instance, one estimates the potential value on improving the model, and the other estimates how easily the model can fully exploit the potential value. Then by alternately optimizing the two weights, valuable and easy instances are selected. Further, when the model becomes stronger after more labels queried, harder instances can be gradually involved by increasing a pace parameter.

Experiments are performed on 9 datasets to validate the proposed approach. Results comparing with state-of-the-art methods show that considering the easiness of instances can boost the performance of active learning.

The rest of the paper is organized as follows. We first review related work in the following section, then the SPAL

method is proposed next, followed by the experimental study. And at last we conclude this work.

## Related Work

There are many selection criteria proposed for active learning over the past few decades. Miscellaneous criteria evaluate how useful an instance is for improving the model from different aspects. Most of them estimate the informativeness or representativeness of the instance. The former has been implemented by error reduction (Tang et al. 2012), query by committee (Seung, Opper, and Sompolinsky 1992), uncertainty (Yan and Huang 2018), etc. While the latter has been implemented by clustering (Dasgupta and Hsu 2008) or density estimation (Zhu et al. 2010). Recently, it has been validated that simultaneously considering both the informativeness and representativeness is usually superior to using a single criterion alone (Wang and Ye 2013; Huang and Zhou 2013; Huang, Jin, and Zhou 2014). Wang and ye (2013) optimize a well-designed objective function, which consists of a term estimating the uncertainty and a term estimating the distribution difference between labeled set and the whole data set. Huang et al. (2014) consider the min-max view of active learning (Hoi et al. 2008), and exploit both informativeness and representativeness with the help of unlabeled data in the semi-supervised learning setting. While all these methods try to estimate the potential value of an instance for improving the model, they neglect whether the selected example can be fully utilized by the current model.

Learning concepts from easy to hard was first proposed in Curriculum learning (Bengio et al. 2009), in which the 'curriculum' is defined intuitively by human. Self-paced learning reformulates this learning process as an optimization problem in order to make it more implementable. This algorithm alternately optimizes model and sample weights with a gradually increasing pace parameter, and sequentially involves instances from easy to hard. In the past few years, SPL has yielded brilliant results in many applications, such as visual category discovery (Lee and Grauman 2011), long-term tracking (Supancic and Ramanan 2013), multi-view clustering (Xu, Tao, and Xu 2015), multi-instance learning (Zhang, Meng, and Han 2017). There are also a few works trying to integrate this paradigm into other algorithms for better performances. For example, Ma et al. (2017) propose self-paced co-training which aggregates SPL and co-training to improve the robustness. In (Wang et al. 2017), authors incorporate SPL with boosting method to deal with the noise sensitive problem.

There is one work trying to combine self-paced learning and active learning for face identification (Lin et al. 2018). This method on one hand employs self-paced learning to select instances with high confidence and assign them with pseudo labels predicted by the model; and on the other hand employs active learning to select most uncertain instances, and query their ground-truth labels from the oracle. However, these two strategies are employed independently, and thus still have the risk that the actively selected instances may not be fully utilized by the current model. Furthermore, the performance of this method heavily depends on the quality of predicted labels, which could be unstable when the model is trained with very limited labeled data.

## The Proposed Method

In this section, we first discuss the objective function of the proposed method and explain each term in the formula. Then, the optimization strategy is introduced next. And at last we present the steps of the algorithm.

We denote by $D$ the dataset with $n$ instances, which includes a small labeled set $L = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_l}$ with $n_l$ instances, and a large unlabeled set $U = \{\boldsymbol{x}_j\}_{j=n_l+1}^{n_l+n_u}$ with $n_u$ instances, where $n_l \ll n_u$ and $n = n_l + n_u$. At each iteration of active learning, a small batch of instances $Q = \{\boldsymbol{x}_q\}_{q=1}^{b}$ with size $b$ will be selected from $U$ to query their labels from the oracle.

### The objective function

It has been disclosed by self-paced learning that learning from easy to hard can boost the performance. This implies that at a specific training stage, over-complex instances may be less useful for improving the model, and querying their labels can cause waste of labeling cost. While the informative and representative instances could be easy or hard for the current model, over-complex instances may not be fully utilized even they have high potential value. It is thus important to query the right thing at the right time. On one hand, the selected instances should have high potential value for improving the model; and on the other hand, the potential value can be fully exploited by the current model.

To achieve this goal, we introduce a variable $w_j \in [0, 1]$ for each unlabeled instance $\boldsymbol{x}_j$ to estimate the potential value. Specifically, more informative and representative instance should receive a higher value of $w_j$. In addition, another variable $v_j \in [0, 1]$ is introduced to estimate the easiness of instance $\boldsymbol{x}_j$ for the current model. By employing a regularizer imposed on this weight, easier sample will receive a larger $v_j$. Then the following objective function is proposed to optimize the two variables along with the model:

$$\min_{f,\boldsymbol{w},\boldsymbol{v}} \ell(f, \boldsymbol{w}, \boldsymbol{v}) + \lambda g(\boldsymbol{v}) + \mu h(L \cup Q, U \backslash Q) + \gamma \Omega(f)\,, \tag{1}$$

where $f$ is the learning model. The first term calculates the expected loss after the query, $g(\cdot)$ is a self-paced regularizer to filter out over-complex instances, $h(L \cup Q, U \backslash Q)$ is a function to estimate the distribution difference between labeled and unlabeled data, and $\Omega(f)$ is for controlling the model complexity. In summary, $\ell(\cdot)$, $h(\cdot)$ and $g(\cdot)$ are responsible for informativeness, representativeness and easiness, respectively.

Next, we specify the implementations of $\ell(\cdot)$, $h(\cdot)$ and $g(\cdot)$ in detail. For simplicity, we employ the least squared loss to estimate the expected losses of instances. And we will get the following form of $\ell(\cdot)$:

$$\ell(f, \boldsymbol{w}, \boldsymbol{v}) = \sum_{i=1}^{n_l} (y_i - f(\boldsymbol{x}_i))^2 + \sum_{j=1}^{n_u} v_j w_j (\hat{y_j} - f(\boldsymbol{x}_j))^2. \tag{2}$$

One challenge here is the ground-truth labels of the selected data are unknown before querying. Inspired by (Wang and Ye 2013), we consider the upper bound of the risk by taking $\hat{y_j} = -sign(f(\boldsymbol{x}_j))$ as the pseudo label of $\boldsymbol{x}_j \in U$. Then we have the following expected loss with self-paced weights:

$$\begin{aligned}\ell(f, \boldsymbol{w}, \boldsymbol{v}) = &\sum_{i=1}^{n_l}(y_i - f(\boldsymbol{x}_i))^2 \\ &+ \sum_{j=1}^{n_u} v_j w_j (f(\boldsymbol{x}_j)^2 + 2|f(\boldsymbol{x}_j)| + 1).\end{aligned} \tag{3}$$

Obviously, by optimizing the above formulation, the informative instance with a small $|f(\boldsymbol{x}_j)|$ will receive a large $w_j$. In other words, the uncertain instances will be preferred in the active selection.

Then, with the most representative instances selected, the distributions of labeled and unlabeled data should be close after the query. The model trained on the queried instances is expected to generalize well on the unseen data coming from the same distribution. We implement $h(\cdot)$ based on Maximum Mean Discrepancy (MMD) (Borgwardt et al. 2006; Gretton et al. 2006), which is a commonly used method for estimating the difference between two distributions.

Formally, we have:

$$\begin{aligned}&h(L \cup Q, U \backslash Q) = MMD_\phi^2(L \cup Q, U \setminus Q) \\ =&\left\| \frac{1}{n_l + b} \left( \sum_{\boldsymbol{x}_i \in L} \phi(\boldsymbol{x}_i) + \sum_{\boldsymbol{x}_j \in U} w_j \phi(\boldsymbol{x}_j) \right) \right. \\ &\left. - \frac{1}{n_u - b} \sum_{\boldsymbol{x}_j \in U} (1 - w_j)\phi(\boldsymbol{x}_j) \right\|_{\mathcal{H}}^2,\end{aligned} \tag{4}$$

where $\phi : \mathcal{X} \to \mathcal{H}$ is a mapping from the feature space to the Reproducing Kernel Hilbert Space (RKHS). $w_j$ is served as the indicator variable here which is relaxed to a continuous value $[0, 1]$.

According to the discussion in previous work, $MMD_\phi^2(p, q)$ will vanish if $p = q$. Therefor, our purpose is to ensure the selected instances can lead to a small value for the above formulation.

With the similar derivations in (Chattopadhyay et al. 2012), we can have a more simple formulation for the above problem:

$$\min_{\boldsymbol{w}} h(L \cup Q, U \backslash Q) = \min_{\boldsymbol{w}} \boldsymbol{w}^T K_1 \boldsymbol{w} + \boldsymbol{k}\boldsymbol{w}\,, \tag{5}$$

where

$$\begin{aligned}&K_1 = \frac{1}{2} K_{UU}\,, \\ &\boldsymbol{k} = \frac{n_u - b}{n} \mathbf{1}_{n_l} K_{LU} - \frac{n_l + b}{n} \mathbf{1}_{n_u} K_{UU}\,,\end{aligned}$$

$K$ is the kernel matrix and $K_{AB}$ denotes the sub-matrix of $K$ between set $A$ and $B$. $\mathbf{1}_{n_l}$ and $\mathbf{1}_{n_u}$ are vectors with all elements being 1. By minimizing the above formulation, the representative instance will receive a large $w_j$.

Next we discuss how to implement the self-paced regularizer $g(\cdot)$, whose role is to control the optimization of weight vector $\boldsymbol{v}$ in order to ensure the easy instance can receive a large $v_j$. Here we simply employ the strategy used in (Jiang et al. 2014) as:

$$g(\boldsymbol{v}) = \frac{1}{2}||\boldsymbol{v}||_2^2 - \sum_{j=1}^{n_u} v_j. \tag{6}$$

Note that $\lambda$ in Eq. 1 is the pace parameter. When $\lambda$ is small at early stage, only a small subset of easy examples with small losses will be utilized. With more instances queried, the model becomes stronger, then harder examples can be involved as $\lambda$ iteratively increases during the learning process. It can be shown in the next subsection that by minimizing the above formulation, easy example will receive a large value of $v_j$.

Lastly, with a commonly used $\ell_2$ norm for controlling the model complexity, i.e., $\Omega(f) = ||f||^2$, we can rewrite the objective function in Eq. 1 as follows.

$$\begin{aligned}\min_{f, \boldsymbol{w}, \boldsymbol{v}} &\sum_{i=1}^{n_l}(y_i - f(\boldsymbol{x}_i))^2 + \sum_{j=1}^{n_u}[v_j \cdot w_j(\hat{y_j} - f(\boldsymbol{x}_j))^2 \\ &+ \lambda(\frac{1}{2}v_j^2 - v_j)] + \mu(\boldsymbol{w}^T K_1 \boldsymbol{w} + \boldsymbol{k}\boldsymbol{w}) + \gamma||f||^2 \\ s.t. \quad &w_j \in [0, 1], v_j \in [0, 1] \quad \forall j = 1 \cdots n_u\,.\end{aligned} \tag{7}$$

As a result, we formulate the active selection procedure as a concise optimization problem, which incorporates the easiness, informativeness and representativeness into an unified framework for self-paced active learning. Next, we will discuss the optimizing strategy of our method.

## Optimization

We use alternative optimization strategy (Bezdek and Hathaway 2003) to optimize the objective function in Eq. 7.

**Optimize $f$ with the fixed $\boldsymbol{v}$ and $\boldsymbol{w}$** Firstly, we introduce the method to optimize $f$ with fixed $\boldsymbol{v}$ and $\boldsymbol{w}$. For simplicity, $f$ is implemented with the kernel form $f(\boldsymbol{x}_i) = \sum_{\boldsymbol{x}_k \in L} \theta_k k(\boldsymbol{x}_k, \boldsymbol{x}_i)$, where $k(\cdot)$ is the kernel function. Then the task is to learn $\boldsymbol{\theta}$, which leads to the following optimization problem:

$$\begin{aligned}\min_{\boldsymbol{\theta}} &\sum_{i=1}^{n_l}(y_i - \sum_{\boldsymbol{x}_k \in L} \theta_k k(\boldsymbol{x}_k, \boldsymbol{x}_i))^2 + \\ &\sum_{j=1}^{n_u} \left[ v_j \cdot w_j \Big( \sum_{\boldsymbol{x}_k \in L} \theta_k k(\boldsymbol{x}_k, \boldsymbol{x}_j) \Big)^2 + \right. \\ &\left. 2 v_j \cdot w_j \left| \sum_{\boldsymbol{x}_k \in L} \theta_k k(\boldsymbol{x}_k, \boldsymbol{x}_j) \right| \right] + \gamma \boldsymbol{\theta}^T K_{LL} \boldsymbol{\theta}\,.\end{aligned} \tag{8}$$

The alternating direction method of multipliers (ADMM) (Boyd et al. 2011) is employed to solve this problem. There are mainly three key steps when performing ADMM to solve Eq. 8. Firstly, we construct

auxiliary variable $\boldsymbol{z}$. Then, the augmented Lagrangian for the original function is constructed. Finally, we optimize the original variable $\boldsymbol{\theta}$, auxiliary variable $\boldsymbol{z}$, and the dual variable $\boldsymbol{\delta}$ in augmented Lagrangian alternately. Following we discuss the three steps in detail.

For the auxiliary variable, we let $z_j = \sum_{\boldsymbol{x}_k \in L} \theta_k k(\boldsymbol{x}_k, \boldsymbol{x}_j)$ for each $\boldsymbol{x}_j \in U$. Note that we filter out some less important samples whose weight $w_j \cdot v_j$ is less than a specified small threshold for efficiency in optimizing $\boldsymbol{\theta}$. Then the optimization problem can be rewritten as:

$$\begin{aligned} \min_{\boldsymbol{\theta}} & \sum_{i=1}^{n_l} (y_i - \sum_{\boldsymbol{x}_k \in L} \theta_k k(\boldsymbol{x}_k, \boldsymbol{x}_i))^2 + \gamma \boldsymbol{\theta}^T K_{LL} \boldsymbol{\theta} \\ & + \sum_{j=1}^{n_u} \left[ v_j \cdot w_j (z_j)^2 + 2 v_j \cdot w_j |z_j| \right] \\ s.t. \quad & z_j - \sum_{\boldsymbol{x}_k \in L} \theta_k k(\boldsymbol{x}_k, \boldsymbol{x}_j) = 0 \quad \forall j = 1 \cdots n_u \,. \end{aligned} \tag{9}$$

The augmented Lagrangian is:

$$\begin{aligned} & \sum_{i=1}^{n_l} (y_i - \sum_{\boldsymbol{x}_k \in L} \theta_k k(\boldsymbol{x}_k, \boldsymbol{x}_i))^2 + \sum_{j=1}^{n_u} \Big[ v_j \cdot w_j (z_j)^2 \\ & + 2 v_j \cdot w_j |z_j| + \delta_j (z_j - \sum_{\boldsymbol{x}_k \in L} \theta_k k(\boldsymbol{x}_k, \boldsymbol{x}_j)) \\ & + \frac{\rho}{2} (z_j - \sum_{\boldsymbol{x}_k \in L} \theta_k k(\boldsymbol{x}_k, \boldsymbol{x}_j))^2 \Big] + \gamma \boldsymbol{\theta}^T K_{LL} \boldsymbol{\theta} \,, \end{aligned} \tag{10}$$

where $\rho$ is a parameter in ADMM.

Finally, by denoting $\circ$ as the element-wise product of vectors, $(\cdot)_+$ as setting the negative entries of the argument vector to 0, $\boldsymbol{y_l} = [y_1, \cdots, y_{n_l}]^T$, $\boldsymbol{\eta} = \boldsymbol{v} \circ \boldsymbol{w}$, $\boldsymbol{\epsilon} = [\epsilon_1, \cdots, \epsilon_{n_u}]^T$, and $\epsilon_j = \sqrt{\eta_j + \frac{\rho}{2}}$. Then we can get the following updating rules:

$$\begin{aligned} \boldsymbol{\theta}^{k+1} &= A^{-1} \boldsymbol{r}^T \,, \\ \boldsymbol{z}^{k+1} &= diag(\boldsymbol{\epsilon})^{-1} \boldsymbol{\zeta}, \\ \boldsymbol{\delta}^{k+1} &= \boldsymbol{\delta}^k + \rho (\boldsymbol{z}^{k+1} - K_{LU}^T (\boldsymbol{\theta}^{k+1})) \,, \end{aligned} \tag{11}$$

where

$$\begin{aligned} A =& K_{LL} K_{LL}^T + \frac{\rho}{2} K_{LU} K_{LU}^T + \gamma K_{LL}, \\ \boldsymbol{r} =& \boldsymbol{y_l}^T K_{LL}^T + \frac{1}{2} \boldsymbol{\delta}^{kT} K_{LU}^T + \frac{\rho}{2} \boldsymbol{z}^{kT} K_{LU}^T, \\ \boldsymbol{\zeta} =& \arg\min \frac{1}{2} ||\boldsymbol{\zeta} - \boldsymbol{o}||_2^2 + \sum_{j=1}^{n_u} \xi_j |\zeta_j| \\ =& sign(\boldsymbol{o}) \circ (|\boldsymbol{o}| - \boldsymbol{\xi})_+, \\ \boldsymbol{o} =& \frac{1}{2} diag(\boldsymbol{\epsilon})^{-1} (\rho \cdot K_{LU}^T \boldsymbol{\theta}^{k+1} - \boldsymbol{\delta}^k), \\ \boldsymbol{\xi} =& diag(\boldsymbol{\epsilon})^{-1} \boldsymbol{\eta} \,. \end{aligned}$$

**Optimize $\boldsymbol{w}$ with the fixed $f$ and $\boldsymbol{v}$** To optimize $\boldsymbol{w}$ for the fixed $f$ and $\boldsymbol{v}$, Eq. 7 becomes:

$$\begin{aligned} \min_{\boldsymbol{w}^T \boldsymbol{v} = b, \boldsymbol{w} \in [0,1]^{n_u}} & \sum_{j=1}^{n_u} [v_j \cdot w_j (\hat{y_j} - f(x_j))^2] \\ & + \mu (\boldsymbol{w}^T K_1 \boldsymbol{w} + \boldsymbol{k} \boldsymbol{w}) \,. \end{aligned} \tag{12}$$

**Algorithm 1** The SPAL Algorithm

**Input:**
  Training set $L$ and $U$;
**Initializing:**
  Initialize $\boldsymbol{v} = \boldsymbol{1}_{n_u}$, $\boldsymbol{w} = \boldsymbol{1}_{n_u}$;
**Repeat until convergence:**
  Update $f$ by solving Eq. 8 through ADMM;
  Update $\boldsymbol{w}$ by solving Eq. 13;
  Update $\boldsymbol{v}$ by solving Eq. 14;
  $Q \leftarrow$ top $b$ instances of $U$ with largest $v_j \cdot w_j$ values;
  $U = U \setminus Q$;  $L = L \cup Q$;
  Train the model based on $L$.

By denoting $\boldsymbol{c} = \mu \boldsymbol{k} + \boldsymbol{a}^T$, where $a_j = v_j (f(\boldsymbol{x}_j)^2 + 2|f(\boldsymbol{x}_j)|)$, the above function can be further rewritten as:

$$\min_{\boldsymbol{w}^T \boldsymbol{v} = b, \boldsymbol{w} \in [0,1]^{n_u}} \boldsymbol{w}^T (\mu K_1) \boldsymbol{w} + \boldsymbol{c} \boldsymbol{w} \,. \tag{13}$$

This is a quadratic programming problem, and can be efficiently solved with existing toolbox.

**Optimize $\boldsymbol{v}$ with the fixed $f$ and $\boldsymbol{w}$** Finally, when optimizing $\boldsymbol{v}$ with fixed $f$ and $\boldsymbol{w}$, we have the following problem:

$$\min_{\boldsymbol{v} \in [0,1]^{n_u}} \sum_{j=1}^{n_u} \left[ v_j \tilde{\ell}_j + \lambda (\frac{1}{2} v_j^2 - v_j) \right] , \tag{14}$$

where

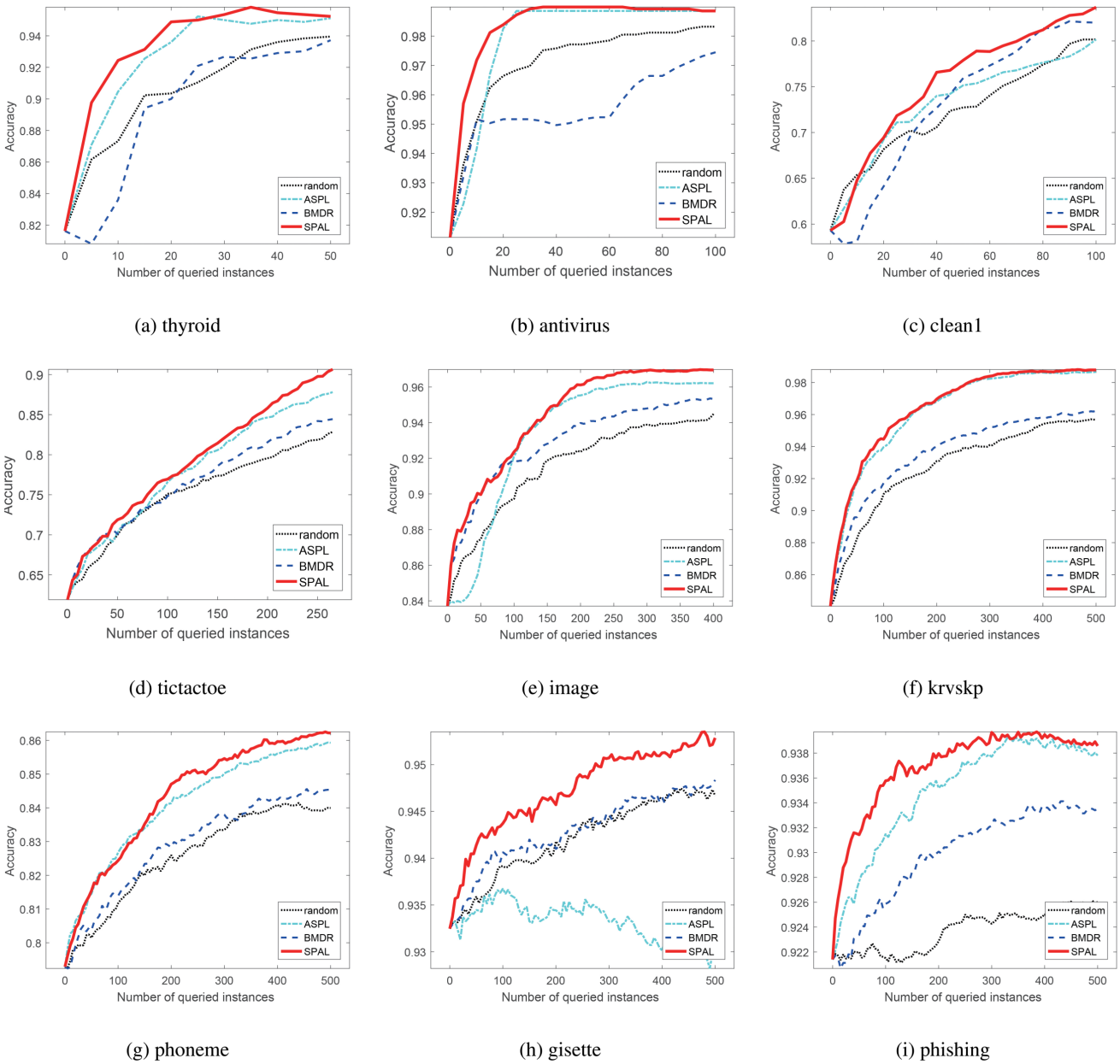
$$\tilde{\ell}_j = w_j (\hat{y_j} - f(\boldsymbol{x}_j))^2 \quad \forall j = 1 \cdots n_u.$$

With linear soft weighting regularizer $g(\boldsymbol{v})$, this problem has the closed form solution for $v_j$:

$$v_j^* = \begin{cases} -\frac{\tilde{\ell}_j}{\lambda} + 1 & \tilde{\ell}_j < \lambda \\ 0 & \tilde{\ell}_j \geq \lambda \,. \end{cases} \tag{15}$$

It can be observed that, The weight $\boldsymbol{v}$ is updated based on the current losses of instances. By adopting the self-paced regularizer $g(\boldsymbol{v})$, the solution of $v_j$ is inversely proportional to its weighted loss $\tilde{\ell}_j$. Thus the easily learned samples with smaller losses can receive higher value of $v_j$. The pace parameter $\lambda$ can be taken as the threshold to filter out over-complex instances. Note that when the pace parameter $\lambda = \infty$, all entries of $\boldsymbol{v}$ will be 1; at this point, our method will degenerate to the active learning approach that does not consider the easiness.

We summarize the framework of SPAL in Algorithm 1. At each iteration, $f$, $\boldsymbol{w}$ and $\boldsymbol{v}$ will be optimized alternately until converge. Instance with high potential value can be identified by the optimized $w_j$, while easy instance for the current model will receive a large $v_j$. We thus select the instances with the largest $v_j \cdot w_j$ to ensure they not only have high potential value for improving the model, but also can be fully utilized by the current model. After updating the model with $L \cup Q$, we evaluate the performance on the test set.




(a) thyroid (b) antivirus (c) clean1

(d) tictactoe (e) image (f) krvskp

(g) phoneme (h) gisette (i) phishing

Figure 1: Performance comparison.

## Experiments

In this section, we first introduce the compared methods and datasets in the experiments, followed by the implementation settings. Then we illustrate and analyze the performance comparison results with other methods. Finally, the experiment with different ratios of initially labeled data is performed to examine the robustness of our method.

### Settings

To validate the effectiveness of our approach, we conduct experiments to compare the following methods:

- **ASPL:** Select a batch of most uncertain instances to query and a batch of high confidence samples to assign pseudo-labels (Lin et al. 2018).
- **BMDR:** Select a batch of informative and representative instances by optimizing the ERM risk bound for active learning (Wang and Ye 2013).
- **Random:** Select a batch of instances randomly.
- **SPAL:** The method proposed in this paper.

We perform the experiments on 9 datasets, whose sizes are summarized in Table 1. For each dataset, we randomly

Table 1: Datasets used in the experiments.

| Dataset | thyroid | antivirus | clean1 |
|---|---|---|---|
| # Instances | 215 | 373 | 476 |
| Dataset | tictactoe | image | krvskp |
| # Instances | 958 | 2086 | 3196 |
| Dataset | phoneme | gisette | phishing |
| # Instances | 5404 | 7000 | 11055 |

sample 40% instances as the test set, and the rest 60% instances for the training. Further, 5% of the training set is used as the initially labeled data, while the rest instances consist of the unlabeled pool for active selection. The data partition is repeated randomly for 10 times. We fix batch size $b = 5$ for all methods.

Note that the ASPL will add two batches of instances into $L$ in each iteration, with half from querying and half from prediction. This causes that the end point of ASPL is earlier than others. Thus we also stop other methods to ensure the numbers of queried instances are the same. For the relatively large datasets, we report the performances of early stage to demonstrate that at a specific training stage, over-complex examples may be less useful than easy ones for improving the model. It is thus important to query the right thing at the right time.

The parameters of BMDR are set to the recommended values in their paper. Specifically, the regularized weight $\gamma = 0.1$ and the trade-off parameter $\mu = 1000$. For ASPL, it targets for specific application and can not be applied to binary classification problem directly, so we simplify it to select two batches of samples with the same batch size, one is the most uncertain instances for querying and the other is the most confident instances for assigning predicted labels. For the proposed method SPAL, we fix $\mu = 0.1$, and $\gamma = 0.1$. For the SPL parameter $\lambda$, we initialize it with a certain value which is selected from $\{0.1, 0.01\}$, and follow the method used in (Lin et al. 2018) to update it linearly with a small fixed value. In our experiments, we fix $\lambda_{pace} = 0.01$ for all datasets. Specifically, we have the following updating rule for $\lambda$ at $t$th iteration:

$$\lambda_t = \lambda_{initial} + (t-1) * \lambda_{pace} .$$

CVX (Grant and Boyd 2014) and MOSEK [1] are used to solve the QP problem. We follow (Wang and Ye 2013) to employ a regularized linear model to implement the classification model for all methods.

## Performance comparison

We plot the average accuracy curves of the proposed SPAL and compared methods with queried instances increasing in Figure 1. To further validate the significance of our method, we also conduct paired t-tests at 95 percent significance level when 20%, 40%, 60%, 80%, 100% of the preset number of queries is reached. We present the win/tie/loss counts of SPAL versus the other methods in Table 2.

[1] http://www.mosek.com/

Table 2: Win/Tie/Loss counts of SPAL versus the other methods with 20%, 40%, 60%, 80%, 100% of the preset number of queries based on paired t-tests at 95 percent significance level.

| Dataset | SPAL versus | | | In All |
|---|---|---|---|---|
| | Random | BMDR | ASPL | |
| thyroid | 4/1/0 | 5/0/0 | 2/3/0 | 11/4/0 |
| antivirus | 4/1/0 | 5/0/0 | 0/5/0 | 9/6/0 |
| clean1 | 4/1/0 | 1/4/0 | 3/2/0 | 8/7/0 |
| tictactoe | 4/1/0 | 4/1/0 | 1/4/0 | 9/6/0 |
| image | 4/1/0 | 4/1/0 | 4/1/0 | 12/3/0 |
| krvskp | 5/0/0 | 5/0/0 | 2/3/0 | 12/3/0 |
| phoneme | 5/0/0 | 4/1/0 | 0/5/0 | 9/6/0 |
| gisette | 5/0/0 | 3/2/0 | 5/0/0 | 13/2/0 |
| phishing | 5/0/0 | 5/0/0 | 1/4/0 | 11/4/0 |
| In All | 40/5/0 | 36/9/0 | 18/27/0 | 94/41/0 |

It can be observed from the figure that the proposed SPAL approach outperforms the other methods in most cases. When comparing with BMDR, our method is always superior. It implies that considering the easiness of the instances can save the labeling cost by filtering out instances that are over complex for the classification model. ASPL works well on some datasets but fails on the others. Note that in addition to the queried instances, ASPL also adds a batch of instances with predicted labels. That is why its performance is less stable. Because the predicted labels could be unreliable when the model is not well trained. As expected, the random strategy is usually the worst one.

Table 2 shows that our method can outperform the baseline methods significantly in most cases. Note that, although ASPL achieves better performance than random and BMDR by using extra self-annotated instances in model training, it still has the risk that the labeled instances may not be fully utilized by the model. We believe this is the reason why our method can outperform the others.

## Study on different initially labeled ratios

In this subsection, we further perform the experiments with different ratios of initially labeled data to examine the performances of compared approaches. Specifically, we compare the methods when the 1%, 5%, 10% and 20% of the training set is initially labeled while other settings remain unchanged. Because of the space limitation, we report the average value of the accuracy curve instead of plotting the whole curve. For each case, the best result and its comparable performances are highlighted in boldface based on paired t-tests at 95 percent significance level. The mean and standard deviation of accuracies are presented in Table 3.

We can observe that SPAL achieves the best performance for most cases, and for the few cases that our method is not the best, it is comparable to the best performance. These results imply that our method is rather stable and can outperform the others with different ratios of initially labeled data. Table 3 shows that ASPL method prefers larger initially labeled ratio. Note that ASPL uses more training data than

Table 3: Influence of different initially labeled ratios (mean ± std). The best performance and its comparable performances based on paired t-tests at 95 percent significance level are highlighted in boldface.

| Dataset | Different labeled ratios | Methods | | | |
|---|---|---|---|---|---|
| | | SPAL | Random | BMDR | ASPL |
| thyroid | 1% | $\mathbf{0.879 \pm 0.019}$ | $0.854 \pm 0.020$ | $0.837 \pm 0.016$ | $0.783 \pm 0.137$ |
| | 5% | $\mathbf{0.929 \pm 0.016}$ | $0.899 \pm 0.035$ | $0.889 \pm 0.032$ | $0.920 \pm 0.015$ |
| | 10% | $\mathbf{0.933 \pm 0.014}$ | $0.913 \pm 0.030$ | $0.916 \pm 0.023$ | $\mathbf{0.931 \pm 0.018}$ |
| | 20% | $\mathbf{0.935 \pm 0.013}$ | $\mathbf{0.929 \pm 0.022}$ | $0.928 \pm 0.015$ | $\mathbf{0.938 \pm 0.013}$ |
| antivirus | 1% | $\mathbf{0.973 \pm 0.009}$ | $0.956 \pm 0.013$ | $0.921 \pm 0.025$ | $0.964 \pm 0.009$ |
| | 5% | $\mathbf{0.982 \pm 0.007}$ | $0.970 \pm 0.011$ | $0.954 \pm 0.017$ | $0.978 \pm 0.008$ |
| | 10% | $\mathbf{0.985 \pm 0.007}$ | $0.976 \pm 0.012$ | $0.963 \pm 0.018$ | $\mathbf{0.984 \pm 0.009}$ |
| | 20% | $\mathbf{0.987 \pm 0.008}$ | $0.980 \pm 0.011$ | $0.976 \pm 0.013$ | $\mathbf{0.986 \pm 0.008}$ |
| clean1 | 1% | $\mathbf{0.708 \pm 0.030}$ | $0.687 \pm 0.031$ | $0.681 \pm 0.025$ | $0.654 \pm 0.026$ |
| | 5% | $\mathbf{0.749 \pm 0.034}$ | $0.719 \pm 0.032$ | $\mathbf{0.725 \pm 0.022}$ | $0.727 \pm 0.038$ |
| | 10% | $\mathbf{0.768 \pm 0.036}$ | $0.742 \pm 0.029$ | $0.739 \pm 0.028$ | $\mathbf{0.760 \pm 0.026}$ |
| | 20% | $\mathbf{0.788 \pm 0.024}$ | $\mathbf{0.775 \pm 0.025}$ | $\mathbf{0.776 \pm 0.020}$ | $0.780 \pm 0.028$ |
| tictactoe | 1% | $\mathbf{0.763 \pm 0.013}$ | $0.727 \pm 0.019$ | $0.722 \pm 0.024$ | $\mathbf{0.756 \pm 0.027}$ |
| | 5% | $\mathbf{0.786 \pm 0.017}$ | $0.748 \pm 0.021$ | $0.761 \pm 0.022$ | $\mathbf{0.775 \pm 0.020}$ |
| | 10% | $\mathbf{0.810 \pm 0.015}$ | $0.766 \pm 0.024$ | $0.749 \pm 0.027$ | $\mathbf{0.803 \pm 0.011}$ |
| | 20% | $\mathbf{0.839 \pm 0.010}$ | $0.794 \pm 0.027$ | $0.769 \pm 0.028$ | $\mathbf{0.838 \pm 0.013}$ |
| image | 1% | $\mathbf{0.933 \pm 0.007}$ | $0.896 \pm 0.011$ | $0.916 \pm 0.007$ | $0.909 \pm 0.013$ |
| | 5% | $\mathbf{0.944 \pm 0.008}$ | $0.916 \pm 0.009$ | $0.929 \pm 0.006$ | $0.933 \pm 0.010$ |
| | 10% | $\mathbf{0.953 \pm 0.007}$ | $0.928 \pm 0.009$ | $0.938 \pm 0.005$ | $0.949 \pm 0.007$ |
| | 20% | $\mathbf{0.961 \pm 0.004}$ | $0.941 \pm 0.007$ | $0.947 \pm 0.004$ | $\mathbf{0.960 \pm 0.005}$ |
| krvskp | 1% | $\mathbf{0.942 \pm 0.004}$ | $0.899 \pm 0.012$ | $0.910 \pm 0.005$ | $0.936 \pm 0.003$ |
| | 5% | $\mathbf{0.964 \pm 0.004}$ | $0.928 \pm 0.010$ | $0.937 \pm 0.006$ | $0.962 \pm 0.004$ |
| | 10% | $\mathbf{0.974 \pm 0.004}$ | $0.943 \pm 0.009$ | $0.949 \pm 0.008$ | $0.972 \pm 0.004$ |
| | 20% | $\mathbf{0.980 \pm 0.003}$ | $0.956 \pm 0.006$ | $0.958 \pm 0.006$ | $\mathbf{0.979 \pm 0.004}$ |
| phoneme | 1% | $\mathbf{0.829 \pm 0.007}$ | $0.808 \pm 0.008$ | $0.812 \pm 0.009$ | $\mathbf{0.823 \pm 0.012}$ |
| | 5% | $\mathbf{0.844 \pm 0.008}$ | $0.826 \pm 0.008$ | $0.829 \pm 0.006$ | $\mathbf{0.841 \pm 0.007}$ |
| | 10% | $\mathbf{0.851 \pm 0.004}$ | $0.837 \pm 0.007$ | $0.838 \pm 0.008$ | $\mathbf{0.850 \pm 0.005}$ |
| | 20% | $\mathbf{0.861 \pm 0.006}$ | $0.845 \pm 0.007$ | $0.845 \pm 0.007$ | $\mathbf{0.859 \pm 0.005}$ |
| gisette | 1% | $\mathbf{0.945 \pm 0.003}$ | $0.930 \pm 0.005$ | $0.931 \pm 0.005$ | $0.927 \pm 0.003$ |
| | 5% | $\mathbf{0.947 \pm 0.003}$ | $0.942 \pm 0.004$ | $0.943 \pm 0.005$ | $0.933 \pm 0.005$ |
| | 10% | $\mathbf{0.951 \pm 0.004}$ | $0.946 \pm 0.004$ | $0.947 \pm 0.003$ | $0.935 \pm 0.003$ |
| | 20% | $\mathbf{0.952 \pm 0.003}$ | $0.950 \pm 0.003$ | $0.950 \pm 0.004$ | $0.938 \pm 0.003$ |
| phishing | 1% | $\mathbf{0.934 \pm 0.003}$ | $0.919 \pm 0.007$ | $0.918 \pm 0.006$ | $0.931 \pm 0.002$ |
| | 5% | $\mathbf{0.937 \pm 0.003}$ | $0.924 \pm 0.009$ | $0.930 \pm 0.004$ | $0.935 \pm 0.003$ |
| | 10% | $\mathbf{0.936 \pm 0.003}$ | $0.925 \pm 0.008$ | $0.926 \pm 0.008$ | $0.934 \pm 0.003$ |
| | 20% | $\mathbf{0.935 \pm 0.003}$ | $0.927 \pm 0.006$ | $0.927 \pm 0.007$ | $0.932 \pm 0.004$ |

the other approaches, because it adds two batches with one from querying and one from prediction. When there is more labeled data, the model prediction is more reliable, and thus ASPL can benefit more from the extra pseudo labels. For BMDR, its performance is still worse than ours even in different initial ratios of labeled data which implies that it is important to further consider the easiness of instances even they have high potential value.

In addition, we also observe some trends in the table that the proposed method SPAL favors the case with less labeled data. One possible reason is that many examples are over-difficult for a simple model, and thus we need a self-paced strategy to select the easy ones at such an early learning stage to get cost-effective queries. We believe this is an advantage because active learning is especially important when labeled data is limited.

## Conclusion

In this paper, we propose a novel batch mode active learning approach SPAL to query the right thing at the right time. On one hand, informativeness and representativeness are considered such that the selected instances have high potential value for improving the model; on the other hand, easiness is exploited to make sure the potential value can be fully utilized by the model. These two aspects are incorporated into an unified framework of self-paced active learning. Experiments show that, our method is superior to the state-of-the-art batch mode active learning methods. In the future, we plan to further examine the effectiveness of the proposed framework when the easiness of instances are known.

## References


Basu, S., and Christensen, J. 2013. Teaching classification boundaries to humans. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 41–48.

Bezdek, J. C., and Hathaway, R. J. 2003. Convergence of alternating optimization. *Neural, Parallel, and Scientific Computations* 11(4):351–368.

Borgwardt, K. M.; Gretton, A.; Rasch, M. J.; Kriegel, H.; Schölkopf, B.; and Smola, A. J. 2006. Integrating structured biological data by kernel maximum mean discrepancy. In *Proceedings 14th International Conference on Intelligent Systems for Molecular Biology*, 49–57.

Boyd, S. P.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3(1):1–122.

Chattopadhyay, R.; Wang, Z.; Fan, W.; Davidson, I.; Panchanathan, S.; and Ye, J. 2012. Batch mode active sampling based on marginal probability distribution matching. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 741–749.

Dasgupta, S., and Hsu, D. J. 2008. Hierarchical sampling for active learning. In *Proceedings of the 15th International Conference on Machine Learning*, 208–215.

Grant, M., and Boyd, S. 2014. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. J. 2006. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems*, 513–520.

Hoi, S. C. H.; Jin, R.; Zhu, J.; and Lyu, M. R. 2008. Semi-supervised SVM batch mode active learning for image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Huang, S., and Zhou, Z. 2013. Active query driven by uncertainty and diversity for incremental multi-label learning. In *IEEE 13th International Conference on Data Mining*, 1079–1084.

Huang, S.; Jin, R.; and Zhou, Z. 2014. Active learning by querying informative and representative examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(10):1936–1949.

Jiang, L.; Meng, D.; Mitamura, T.; and Hauptmann, A. G. 2014. Easy samples first: Self-paced reranking for zero-example multimedia search. In *Proceedings of the ACM International Conference on Multimedia*, 547–556.

Khan, F.; Mutlu, B.; and Zhu, X. 2011. How do humans teach: On curriculum learning and teaching dimension. In *Advances in Neural Information Processing Systems*, 1449–1457.

Kumar, M. P.; Packer, B.; and Koller, D. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, 1189–1197.

Lee, Y. J., and Grauman, K. 2011. Learning the easy things first: Self-paced visual category discovery. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition*, 1721–1728.

Lin, L.; Wang, K.; Meng, D.; Zuo, W.; and Zhang, L. 2018. Active self-paced learning for cost-effective and progressive face identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40(1):7–19.

Ma, F.; Meng, D.; Xie, Q.; Li, Z.; and Dong, X. 2017. Self-paced co-training. In *Proceedings of the 34th International Conference on Machine Learning*, 2275–2284.

Settles, B. 2009. Active learning literature survey. Technical report, University of Wisconsin-Madison.

Seung, H. S.; Opper, M.; and Sompolinsky, H. 1992. Query by committee. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, 287–294.

Supancic, J. S., and Ramanan, D. 2013. Self-paced learning for long-term tracking. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2379–2386.

Tang, J.; Zha, Z.; Tao, D.; and Chua, T. 2012. Semantic-gap-oriented active learning for multilabel image annotation. *IEEE Transactions on Image Processing* 21(4):2354–2360.

Tang, Y.; Yang, Y.; and Gao, Y. 2012. Self-paced dictionary learning for image classification. In *Proceedings of the 20th ACM Multimedia Conference*, 833–836.

Wang, Z., and Ye, J. 2013. Querying discriminative and representative samples for batch mode active learning. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 158–166.

Wang, K.; Wang, Y.; Zhao, Q.; Meng, D.; and Xu, Z. 2017. SPLBoost: An improved robust boosting algorithm based on self-paced learning. *arXiv preprint arXiv:1706.06341*.

Xu, C.; Tao, D.; and Xu, C. 2015. Multi-view self-paced learning for clustering. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 3974–3980.

Yan, Y., and Huang, S. 2018. Cost-effective active learning for hierarchical multi-label classification. In *Proceedings of the27th International Joint Conference on Artificial Intelligence*, 2962–2968.

Zhang, D.; Meng, D.; Zhao, L.; and Han, J. 2016. Bridging saliency detection to weakly supervised object detection based on self-paced curriculum learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 3538–3544.

Zhang, D.; Meng, D.; and Han, J. 2017. Co-saliency detection via a self-paced multiple-instance learning framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(5):865–878.

Zhu, J.; Wang, H.; Tsou, B. K.; and Ma, M. Y. 2010. Active learning with sampling by uncertainty and density for data annotations. *IEEE Transactions on Audio, Speech & Language Processing* 18(6):1323–1331.