Hindawi
Complexity
Volume 2019, Article ID 8127869, 10 pages
https://doi.org/10.1155/2019/8127869

*Research Article*

# Cognitive Driven Multilayer Self-Paced Learning with Misclassified Samples

**Qi Zhu,[1,2] Ning Yuan,[1] and Donghai Guan[1]**

[1]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*
[2]*Corroborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210093, China*

Correspondence should be addressed to Donghai Guan; dhguan@nuaa.edu.cn

Received 29 November 2018; Accepted 12 February 2019; Published 10 March 2019

Guest Editor: Ke Deng



In recent years, self-paced learning (SPL) has attracted much attention due to its improvement to nonconvex optimization based machine learning algorithms. As a methodology introduced from human learning, SPL dynamically evaluates the learning difficulty of each sample and provides the weighted learning model against the negative effects from hard-learning samples. In this study, we proposed a cognitive driven SPL method, i.e., retrospective robust self-paced learning (R2SPL), which is inspired by the following two issues in human learning process: the misclassified samples are more impressive in upcoming learning, and the model of the follow-up learning process based on large number of samples can be used to reduce the risk of poor generalization in initial learning phase. We simultaneously estimated the degrees of learning-difficulty and misclassified in each step of SPL and proposed a framework to construct multilevel SPL for improving the robustness of the initial learning phase of SPL. The proposed method can be viewed as a multilayer model and the output of the previous layer can guide constructing robust initialization model of the next layer. The experimental results show that the R2SPL outperforms the conventional self-paced learning models in classification task.

## 1. Introduction

By assigning the samples in a meaningful learning order based on prior knowledge, curriculum learning (CL) [1] provides an easy-to-hard learning process, which makes the model more fits human cognition. To make curriculum learning more practical in dealing with machine learning problems, Kumar et al. [2] adaptively assessed the sample learning difficulty in model and proposed self-paced learning method. Specifically, self-paced algorithm actively and dynamically obtains the initial learning sequence from the original data and gradually increases the hard learning samples during each iteration. However, in curriculum learning, the sample learning course sequence is preset. By predefining or dynamically generating learning sequence, curriculum learning and self-paced learning can avoid main function falling into a bad local optimal solution. Many researchers applied curriculum learning and self-paced learning to some tough pattern recognition problems. In the literature [3], Jiang et al. proposed the self-paced curriculum learning, which not only obtains the dynamic sample sequence in the process of model learning, but also makes use of prior knowledge to avoid overfitting. Zhao et al. [4] applied the nonconvex problem of matrix decomposition, which suppresses effectiveness of the noise and outlier in the data on the model. Meanwhile, they pointed out that the strategy of adaptively selecting easy-learning sample sequences is similar to the process of human cognition. James et al. [5] adopted self-paced learning to SVM and achieved promising results in multimodal data retrieval. Self-paced learning has been introduced to many learning models and shown good performance in many real-world applications.

Self-paced learning seems to challenge the conventional learning methods, like active learning, boost, and transfer learning. In the view of machine learning, these boundary samples, noise samples, and outliers will increase the uncertainty of the model and may make the model generate a bad classification boundary. Therefore, compared to easy-learning samples, hard-learning learning samples have drawn much attention from the conventional model. In our work, we

aim to deal with supervised learning problems, in which easy-learning samples correspond to samples with small loss while hard-learning samples correspond to sample with large loss. In unsupervised learning, easy-learning samples mean the samples that are easy to be determined while hard-learning samples denote the samples that will cause the model to be unstable. In the paper, misclassified samples are denoted as the samples that the product of the predicted value and the label is negative. Typically, in AdaBoost learning, the model trains the classifier by changing the sample distribution based on the misclassified samples of previous iterations [6]. Li et al. [7] applied the sequence of AdaBoost to train classifiers, starting with weak learner and progressively boosted as a strong learner. Active learning is a kind of semisupervised learning, and it chooses to label the most valuable samples for the model. These low-confidence samples that may contain useful information are difficult to be chosen, which requires additional expert knowledge to identify. Tur et al. [8] presented a spoken language understanding method by combining active and semisupervised learning with human-label and automatically labeled data. Huang et al. [9] proposed a systematic framework to simultaneously measure the informativeness and the representativeness of an instance. The informativeness criteria reflects the ability of samples in reducing the uncertainty of model based on the labeled data, while the representativeness measures which samples can well represent the unlabeled data. However, self-paced learning model first considers the easy-learning samples with small prediction loss and gradually adopts hard-learning samples with larger prediction loss to extend the training set. The difference between self-paced learning and transfer learning is that the transfer learning improves the generalization of the model by sharing the models in different tasks [10], while the self-paced learning updates and learns itself to obtain the local optimal solution.

Study [11] pointed out the inherent consistency between human recognition and reinforcement learning. In dealing with a learning problem, humans and other animals utilize a harmonious combination of repeating learning and hierarchical sensory processing systems. In self-paced learning, the initial model is trained insufficiency with a few easy-learning samples, which increases the learning risk of follow-up iteration and even reduces the generalization of the final model. The usual practice of solving the small sample problem contains feature selection [12], regularization, adding artificial samples [13], etc. In order to improve the generalization of the initial model consisting of small samples in self-paced learning, we design the recurrent framework, which uses the model of last self-paced learning iteration to repeatedly construct the initial model. Corresponding to the repeating learning process of humans, if the initial model inherits the property of large sample learning model, the obtained final model may be more robust and discriminative.

Meanwhile, although self-paced learning and some conventional machine learning methods (AdaBoost, active learning and transfer learning) are very different in sample processing, we can still absorb the advantages of these conventional methods into self-paced learning. Specifically, in this paper, we propose retrospective robust self-paced learning (R2SPL). In each iteration of self-paced learning, besides considering easy-learning samples, these misclassified samples of last iteration will also be involved in training the model. For example, if the hard-learning samples (their categories are difficult to determine) in the data are the majority, conventional self-paced learning may not get a good local optimal solution. In this case, our proposed method focuses on both easy-learning samples and misclassified samples in each iteration, which can drive the final mature model be robust and discriminative.

Overall, our main contribution can be summarized as follows:

(i) We introduce these misclassified samples accompanied with easy samples with small loss in each iteration to guide the model becomes more discriminative.

(ii) Retrospective self-paced learning is proposed to improve the robustness of the initialization of self-paced learning.

(iii) Experiments results show the proposed method achieves promising result in classification tasks.

The remainder of this paper is organized as follows. We briefly introduce related works on self-paced learning in Section 2. We propose the robust SPL in Section 3. In Section 4, we conduct the experiments on UCI and ADNI datasets. We provide the conclusion and the future research plan in Section 5.

## 2. Related Work

*2.1. Curriculum Learning and Self-Paced Learning.* In 2009, Bengio et al. [1] proposed a method of imitating children education order which is called curriculum learning. Different from conventional machine learning methods obtained from overall sample learning, in their work, they sorted the samples in a meaningful order and learned the model in several sections. Benefiting from the prior knowledge, curriculum learning can get better results than other machine learning models in some tasks. However, arranging the sample order usually requires expert identification, which increases the difficulty and cost of the model. In addition, the ordered sample sequence is static and lacks flexibility in dealing with new samples or tasks. To alleviate this deficiency, Kumar et al. [2] proposed self-paced learning in 2010. Without any prior knowledge and expert identification, self-paced learning can dynamically assign the samples from easy to difficult based on the fitness between the samples and the model. In multimedia retrieval, Lu et al. [14] proposed self-paced reranking model for multimodal data, and the model made significant progress on both image and video search tasks. Zhou et al. [15] brought the self-paced learning to deep neural network, which can adaptively involve the faithful samples into training process. By analyzing the work mechanism of self-paced learning, Fan et al. [16] proposed a general implicit regularized framework. Since self-paced learning is adopted into many models, the commonality among these models lies in the sample processing. In each iteration,

these models usually pick these high-confidence samples which fit the model better to construct the current model and gradually use the remaining low-confidence samples to fine-tune the model to make it become more generalization.

Curriculum learning is the first attempt to combine human cognition sequence and machine learning model. Although curriculum learning has some drawbacks, it brings the idea of easy-to-hard learning to the latter models. Self-paced learning is the extension of curriculum learning, which is more flexible and concise. Similar to human learning, self-paced learning trains samples from easy to difficult and gradually improves the robustness of the model.

*2.2. Tough Samples Learning.* In the sample processing strategy, self-paced learning method is different with some tough samples focused learning methods, like AdaBoost, active learning, and transfer learning. In our work, we try to finely distinguish different types of samples, including easy-learning samples, hard-learning samples, and misclassified samples, and give them different weights in model. By combining the simple classifiers, AdaBoost can deal with complicated problem. For example, in many multiclass problems [17, 18], the distribution of samples is highly complex [19]. Like SVM, AdaBoost can asymptotically achieve a margin distribution which is robust to noise [7, 20, 21]. Active learning is a semisupervised model that uses the unlabeled samples to improve the model obtained by labeled samples. However, since the unlabeled samples have no tags, some data that are difficult to distinguish the types usually need to manually annotate. Otherwise, if these data are identified by the model, it may increase the uncertainty of the model. Lin et al. [22] proposed active self-paced learning that used the characteristic of these two models to automatically annotate the high-confidence and low-confidence samples and incorporated them into training under weak expert recertification. Kumar et al. [2] pointed out that certainty does not imply correctness. Many researchers performed SVM and active learning in some practical applications, like text classification [23, 24], image retrieval [25], and segmentation of images [26]. The model will adjust the weight of the data from original domain, which increases the similarity of the data between target domain and source domain [10]. In the process of children learning, some problems share a common underlying structure but differ in surface manifestations, which is similar to the characteristics of transfer learning [27]. In order to make the models close to human wisdom, many researchers combine the models with environmental feedback and transfer learning [28–30].

In our work, we will focus on both the easy-learning samples and the tough samples, which improve the discrimination of self-paced learning. In each iteration, we will simultaneously select these easy-learning samples and misclassified samples to train. Like human cognition, it is beneficial to improve the generalization of the final self-paced model by simultaneously learning the high-confidence samples and low-confidence samples in each iteration.

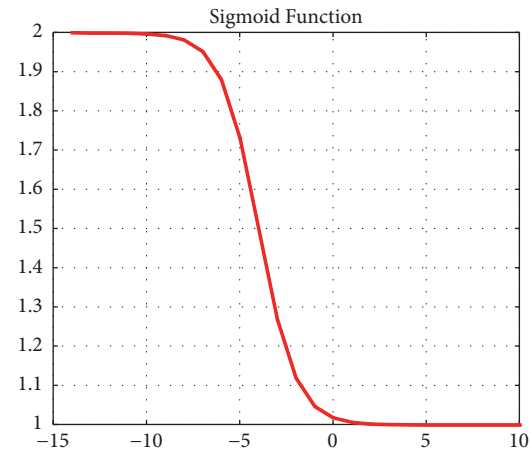


Figure 1: The weight function for weight matrix $Q$.

## 3. Proposed Method

*3.1. Robust SPL.* Specifically, we define a diagonal weight matrix $Q \in \mathbb{R}^{n\times n}$ to denote misclassified weight of each sample. Let $y_i$ and $X_i^T\omega$ represent the label and predicted value of i-th sample, respectively. For binary classification problem, if $y_i X_i^T \omega^{(t)} > 0$, the i-th sample is corrected classified. Otherwise, this sample is considered as misclassified sample. In our work, the weight of these misclassified samples ($y_i X_i^T \omega^{(t)} < 0$) in weight matrix Q should be larger than these corrected samples, and the scope of this type of weight should not vary greatly. Therefore, in our work, we adopt sigmoid function, shown in Figure 1, as weight function with respect to the product of label and predicted value. Given the label vector $y \in \mathbb{R}^{1\times n}$, data matrix $X \in \mathbb{R}^{d\times n}$, and current model parameter $\omega^t \in \mathbb{R}^{d\times 1}$, the misclassified weight of i-th sample can be calculated as

$$q_{ii}^{(t+1)} = c_1 - \frac{1}{1 + \exp - \left(y_i X_i^T \omega^{(t)} + c_2\right)} \tag{1}$$

For supervised problem, self-paced learning function f(V; k) assigns weight to samples based on the sample loss. Those samples with small loss will be viewed as easy-learning samples. However, our model simultaneously considers easy-learning samples and tough samples in each iteration. Specifically, we combine Q and self-paced weight matrix V linearly. Then, the model can be formulated as

$$\begin{aligned} \text{as } L(\omega, V, Q) = \min_{\omega,V} \frac{1}{2} \left\| \left(y - \omega^T X\right)(V + Q)^{1/2} \right\|_2^2 \\ + \frac{\lambda}{2} \|\omega\|_p^p + f(V; k) \end{aligned} \tag{2}$$

where $\lambda$ is the regularization term. To embed structure information in feature extraction, we adopt p-norm on the regression coefficient $\omega$. The closer the value of p gets to 0, the sparser the result of the feature extraction is. f(V; k) is the self-paced weight function and k controls the number of samples which is considered to construct the model. At first, only a

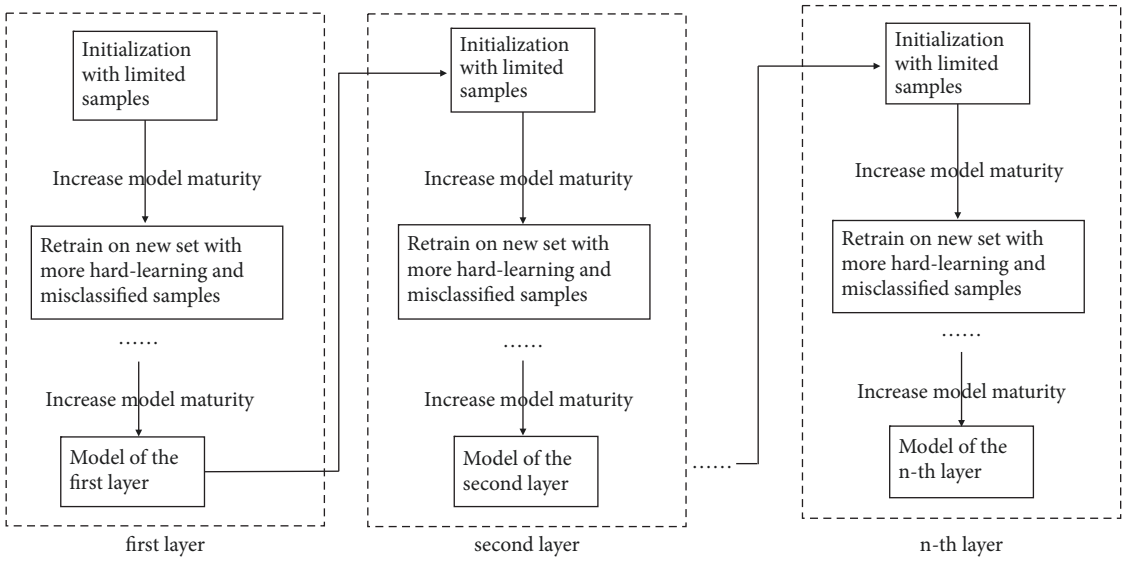


Figure 2: The network model of the proposed method.

few of samples with small loss can be utilized to construct the model. With the decrease of k, more and more samples with larger loss will join the model training process.

Whatever the forms of self-paced function $f(V; k)$ are, they should satisfy three properties [3, 14]: (1) $f(v; k)$ is convex with respect to $v \in [0, 1]$; (2) the sample weight should be monotonically decreasing with respect to its corresponding loss; (3) the sample weight should be monotonically decreasing with respect to the pace parameter k.

Meanwhile, in the process of human cognition, people usually make mistakes due to the lack of knowledge. By constantly summarizing unfamiliar and misunderstand concepts, people can form a more robust knowledge system. Notably, if the children get the help with adults (like teachers and parents) in the process of cognition, they can construct the knowledge framework more rapidly and soundly. The self-paced learning is similar to the education process of children without the help of adults. Therefore, the learned initial model may be not robust enough. To alleviate this deficiency, in this paper, we proposed retrospective robust self-paced learning. Specifically, we cascade multiple self-paced learning algorithms, which can help to reduce the negative impacts in the initialization of follow-up self-paced learning process due to lack of sample simple size problem. Naturally, in the next self-paced learning process, the learning rate can be speeded up moderately. Repeating the process for several times, we can obtain more robust and discriminative model. The framework of the proposed method can be viewed as a multilayer network shown in Figure 2. Firstly, we construct the initial model based on these easy-learning samples. The obtained initial model does not have good discriminability due to lack of sample training. Then, we adopt more hard-learning and misclassified samples to retrain our model, which drives our model to be more robust and discriminative. When the training of this layer is finished, the convergence results will be used as the prior knowledge of the next layer model. Repeat this operation until all n-layer models have been trained. Because self-paced learning stage is essentially a layer of the network. Specifically, the output of the first layer can be used to guide choosing of the easy-learning samples in the initialization of the second layer, which can be expected to be more robust than that learn independently.

*3.2. Optimization.* Since the parameters $\omega$, Q, and V are independent with each other, we can fix other parameters when we calculate each of them. In t + 1-th iteration, each parameter can be calculated by the t-th iteration parameters.

$$\begin{aligned} \omega^{t+1} &= \min_{\omega} L\left(\omega, V^t, Q^t\right) \\ V^{t+1} &= \min_{V} L\left(\omega^t, V, Q^t\right) \\ Q^{t+1} &= \min_{Q} L\left(\omega^t, V^t, Q\right) \end{aligned} \tag{3}$$

In t + 1-th iteration, each portion is a convex problem; the optimal solutions of parameters can be achieved. The solutions of $\omega$, Q, and V are presented as follows.

*(1) The Solution of $\omega$*

$$\begin{aligned} L(\omega, V, Q) = \min_{\omega, V} &\frac{1}{2} \left\| \left(y - \omega^T X\right) (V + Q)^{1/2} \right\|_2^2 \\ &+ \frac{\lambda}{2} \|\omega\|_p^p \end{aligned} \tag{4}$$

To simplify the calculation, we convert the second term of (4) to $\lambda/2\mathrm{tr}(\omega^{\mathrm{T}} G \omega)$. $G \in \mathbb{R}^{d \times d}$ is a diagonal matrix and the diagonal elements can be calculated by

$$g_{ii} = \omega_i^{p-2} \tag{5}$$

1: Input:
2: $X_{train}$: source domain data;
3: $X_{test}$: target domain data;
4: k, k′: self-paced parameter; $\lambda$: regularization parameter
5: For each layer of self-paced learning:
6: Initiate parameter $\omega$ by utilizing the result of last self-paced learning layer.
7: Repeating until convergence
8: Update V by Eq. (12);
9: Update Q by Eq. (1);
10: Update $\omega$ by Eq. (8);
11: Train the model based on V and Q.

Algorithm 1: The algorithm of robust self-paced learning (R2SPL).

where $\omega_i$ is the i-th element of $\omega$. Then, (4) can be equivalently formulated as

$$\min_{\omega} \frac{1}{2} tr\left(\left(y - \omega^T X\right)(V+Q)\left(y - \omega^T X\right)^T\right) + \frac{\lambda}{2} tr\left(\omega^T G \omega\right) \tag{6}$$

Get the derivation of $\omega$ in (6) and set it to 0:

$$X(V+Q)\left(X^T\omega - y^T\right) + \lambda G\omega = 0 \tag{7}$$

Then, the optimal solution of $\omega$ is

$$\omega = \left(X(V+Q)X^T + \lambda G\right)^{-1}\left(X(V+Q)y^T\right) \tag{8}$$

In the next iteration, the model will correct its mistake by guiding the regression coefficient $\omega$ based on the parameter Q. Under the influence of the accumulation of V and Q, which corresponds to easy samples with small loss and misdirected samples, our proposed self-paced learning model will be more robust and discriminative than conventional self-paced learning models which only consider easy sample in each iteration.

*(2) The Solution of V*

$$L(\omega, V, Q) = \min_{V} \frac{1}{2}\left\|\left(y - \omega^T X\right)(V+Q)^{1/2}\right\|_2^2 + f(V;k) \tag{9}$$

In our work, we define f(V; k) as

$$f(V;k) = -\zeta \sum_{i=1}^{n} \log\left(v_{ii} + \zeta k\right) \tag{10}$$

where $\zeta = 1/(k' - k)$, $1/k'$ is used to describe the lower bound of sample loss, and $1/k$ is used to describe the upper bound. Meanwhile, $1/k$ also describes the age of the model. In the initial stage, only easy samples with small loss are considered to construct the model. As $1/k$ grows, more and more complicated samples with larger loss will be adopted to the model to make it more mature. The sample weight can be calculated by our self-paced weight function f(V; k). Get the derivation of V in (9) and set it to 0:

$$\ell_i - \frac{\zeta}{v_{ii} + \zeta k} = 0 \tag{11}$$

where $\ell_i$ is the squared loss of i-th sample. Then, the optimal solution of V is given by

$$v_{ii} = \begin{cases} 1 & \ell_i \le \dfrac{1}{k'} \\ 0 & \ell_i \ge \dfrac{1}{k} \\ \dfrac{\zeta}{\ell_i} - k\zeta & otherwise \end{cases} \tag{12}$$

As mentioned above, we adopt retrospective self-paced learning framework to increase the robustness and discrimination of model. Specifically, the step size of k in the current self-paced learning process is smaller than that in the follow-up process. In our work, we set the step size of first self-paced learning layer is 0.1 and gradually increase it in follow-up process. To simplify the calculation, we set the number of layers to 3 in our method.

*(3) The Solution of Q.* The solution of Q can be calculated by (1). In our work, we apply sigmoid function to assign weight value to matrix Q. Using different self-paced weight functions in (10), we can obtain different models. In detail, we adopt three self-paced learning function, binary, linear, and logarithmic. f(V; k) can be formulated as follows.

*(a) Binary*

$$f(V;k) = -\frac{1}{k}\|V\|_1 \tag{13}$$

*(b) Linear*

$$f(V;k) = \frac{1}{k}\left(\frac{1}{2}\|V\|_2^2 - \sum_{i=1}^{n} v_{ii}\right) \tag{14}$$

*(c) Logarithmic*

$$f(V;k) = \sum_{i=1}^{n}\left(\xi v_{ii} - \frac{\xi v_{ii}}{\log \xi}\right) \tag{15}$$

where $\zeta = (k-1)/k$. The solving algorithm of our model is shown in Algorithm 1.

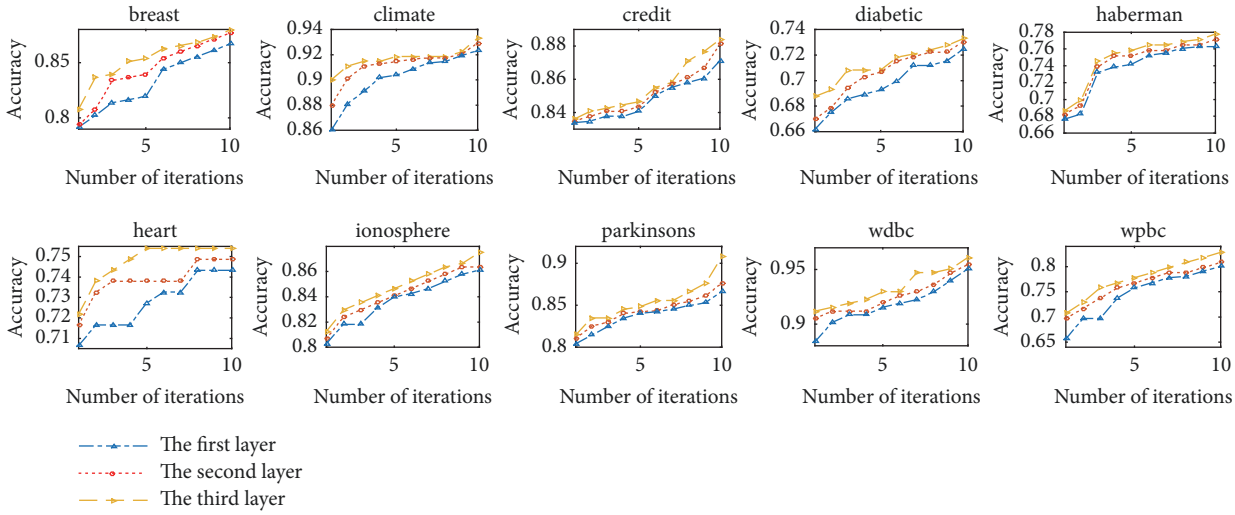


Figure 3: The classification results of the three models on the top ten iterations.

Table 1: Notations of UCI databases.

| Binary-class Dataset | | |
|---|---|---|
| Dataset | Dimension | Number |
| wpbc | 33 | 151/47 |
| wdbc | 30 | 212/357 |
| parkinsons | 22 | 147/48 |
| ionosphere | 225 | 126/99 |
| heart | 22 | 157/110 |
| haberman | 3 | 225/81 |
| diabetic | 19 | 540/611 |
| credit | 15 | 307/383 |
| climate | 18 | 46/494 |
| breast | 9 | 458/241 |

## 4. Experiments

*4.1. Settings.* To evaluate the effectiveness of our proposed method, we conduct our experiment on ten binary classification datasets from UCI repository and Alzheimer's Disease Neuroimaging Initiative (ADNI) database. The detailed information of UCI datasets is presented in Table 1. AD data used in our experiment is obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). In our work, the Alzheimer's Disease (AD) data have 913 samples with 116-dimension, which are consisting of 160 AD patients, 542 MCI patients, and 211 healthy controls (HC). Specifically, the MCI patients can be divided into three stages, 82 Significant Memory Concern (SMC) patients, 273 Early Mild Cognitive Impairment (EMCI) patients, and 187 Late Mild Cognitive Impairment (LMCI) patients. There are five modalities in the ADNI data, including ID (serial number), single nucleotide polymorphism (SNPdata), voxel based morphometry (VBM), fluorodeoxyglucose position emission tomography (FDG), and F-18 florbetapir PET scans amyloid imaging (AV45). In ADNI database, we perform three classification tasks, AD versus HC, MCI versus HC, and SMC versus LMCI. In each classification task, we compare our method with baselines SVM with RBF kernel, AdaBoost and conventional self-paced methods. In the sample processing, AdaBoost adjusts the distribution of training samples based on the performance of basic learners, which makes the misdirected samples in current iteration get more attention in the next iteration.

For conventional self-paced learning models whose weight functions are (13), (14), and (15), we define them as binary, linear, and log for short. Meanwhile, we construct two self-paced learning models based on our proposed models. If the parameter Q is not considered into the retrospective model, we call the model as Easy-SPL for short. When our proposed model is just one level self-pace learning containing parameter Q, it can be defined as Single-SPL. To obtain unbiased results, we adopt 10-fold cross-validation strategy with four measurements, including classification accuracy (ACC), sensitivity (SEN), specificity (SPE), and area under receiver operating characteristic curve (AUC). In UCI databases, we repeat all experiments 30 times with 2-folds cross-validation. In the experiment, we analyze the results of each layer of self-paced leaning process to determine the number of layers. We stop training the model when the convergence results of current self-paced learning process are not significantly improved compared with the previous iteration process. Then, we can determine the number of layers in our model.

*4.2. Experimental Results on UCI and ADNI Data.* At first, we verify the effectiveness of introducing tough samples and retrospective self-paced learning to the model in each iteration on ten UCI datasets. Figure 3 lists the results of each layer of the proposed self-paced learning method. Obviously, after introducing weight matrix Q, the model behaves more discriminative in each iteration. Meanwhile, the model of last self-paced learning process not only has better performance but also behaves more robust than the previous layer. Table 2

Table 2: Results of seven baselines and our model on 10 UCI datasets.

| Dataset | | SVM | AdaBoost | Binary | Linear | Log | Easy-SPL | Single-SPL | R2SPL |
|---|---|---|---|---|---|---|---|---|---|
| wpbc | ACC | 75.08±0.021 | 76.04±0.023 | 76.07±0.008 | 76.33±0.025 | 76.20±0.017 | 76.33±0.009 | 72.86±0.019 | 77.58±0.004 |
| | AUC | 67.10±0.054 | 73.18±0.035 | 74.32±0.063 | 74.22±0.013 | 71.11±0.031 | 50.85±0.019 | 61.00±0.028 | 74.47±0.036 |
| wdbc | ACC | 92.84±0.008 | 93.67±0.021 | 92.88±0.018 | 93.64±0.071 | 93.71±0.044 | 91.34±0.029 | 94.08±0.018 | 94.57±0.006 |
| | AUC | 98.20±0.052 | 98.62±0.021 | 98.19±0.022 | 98.33±0.031 | 98.38±0.007 | 96.79±0.066 | 93.59±0.034 | 98.68±0.055 |
| parkinsons | ACC | 84.67±0.028 | 85.22±0.038 | 85.57±0.037 | 84.23±0.011 | 84.16±0.031 | 77.49±0.041 | 82.61±0.058 | 85.74±0.022 |
| | AUC | 83.13±0.031 | 86.22±0.014 | 84.11±0.053 | 81.89±0.009 | 81.26±0.054 | 83.85±0.071 | 76.47±0.019 | 87.43±0.055 |
| ionosphere | ACC | 85.42±0.029 | 85.97±0.017 | 85.40±0.034 | 84.43±0.061 | 85.23±0.019 | 76.65±0.008 | 87.29±0.011 | 88.37±0.012 |
| | AUC | 96.65±0.057 | 97.14±0.047 | 96.48±0.061 | 96.91±0.031 | 96.77±0.058 | 88.71±0.019 | 83.39±0.022 | 97.23±0.046 |
| heart | ACC | 81.82±0.033 | 82.35±0.031 | 83.96±0.005 | 86.63±0.004 | 87.70±0.061 | 75.40±0.019 | 64.17±0.021 | 89.84±0.023 |
| | AUC | 81.09±0.049 | 80.43±0.012 | 79.84±0.009 | 82.36±0.017 | 83.02±0.032 | 84.22±0.045 | 50.23±0.028 | 85.54±0.046 |
| haberman | ACC | 73.07±0.062 | 73.55±0.019 | 74.20±0.028 | 73.51±0.066 | 73.66±0.057 | 72.79±0.015 | 72.24±0.025 | 74.42±0.022 |
| | AUC | 63.54±0.011 | 67.32±0.015 | 65.91±0.023 | 68.33±0.001 | 68.48±0.068 | 61.47±0.039 | 58.62±0.027 | 68.52±0.003 |
| diabetic | ACC | 68.60±0.029 | 67.59±0.037 | 67.80±0.011 | 69.49±0.016 | 68.76±0.055 | 54.54±0.044 | 63.38±0.021 | 69.91±0.017 |
| | AUC | 76.53±0.022 | 76.39±0.015 | 75.94±0.008 | 76.92±0.064 | 76.37±0.016 | 52.50±0.049 | 63.77±0.054 | 77.34±0.015 |
| credit | ACC | 85.82±0.016 | 85.59±0.068 | 85.35±0.033 | 85.63±0.005 | 85.53±0.051 | 83.09±0.062 | 85.18±0.011 | 85.89±0.033 |
| | AUC | 88.74±0.029 | 89.32±0.014 | 89.46±0.009 | 89.44±0.035 | 88.62±0.026 | 89.03±0.003 | 85.10±0.049 | 89.53±0.061 |
| climate | ACC | 91.41±0.033 | 91.20±0.059 | 91.48±0.064 | 91.25±0.029 | 91.41±0.019 | 91.06±0.006 | 92.33±0.015 | 91.53±0.055 |
| | AUC | 86.79±0.016 | 89.48±0.023 | 83.39±0.055 | 89.75±0.036 | 87.72±0.018 | 89.37±0.049 | 65.68±0.048 | 90.41±0.017 |
| breast | ACC | 96.91±0.039 | 96.82±0.064 | 96.41±0.033 | 96.62±0.019 | 96.45±0.026 | 92.28±0.016 | 94.50±0.017 | 97.13±0.014 |
| | AUC | 99.40±0.023 | 99.43±0.039 | 99.28±0.008 | 99.28±0.046 | 99.29±0.022 | 98.66±0.033 | 93.63±0.028 | 99.47±0.011 |

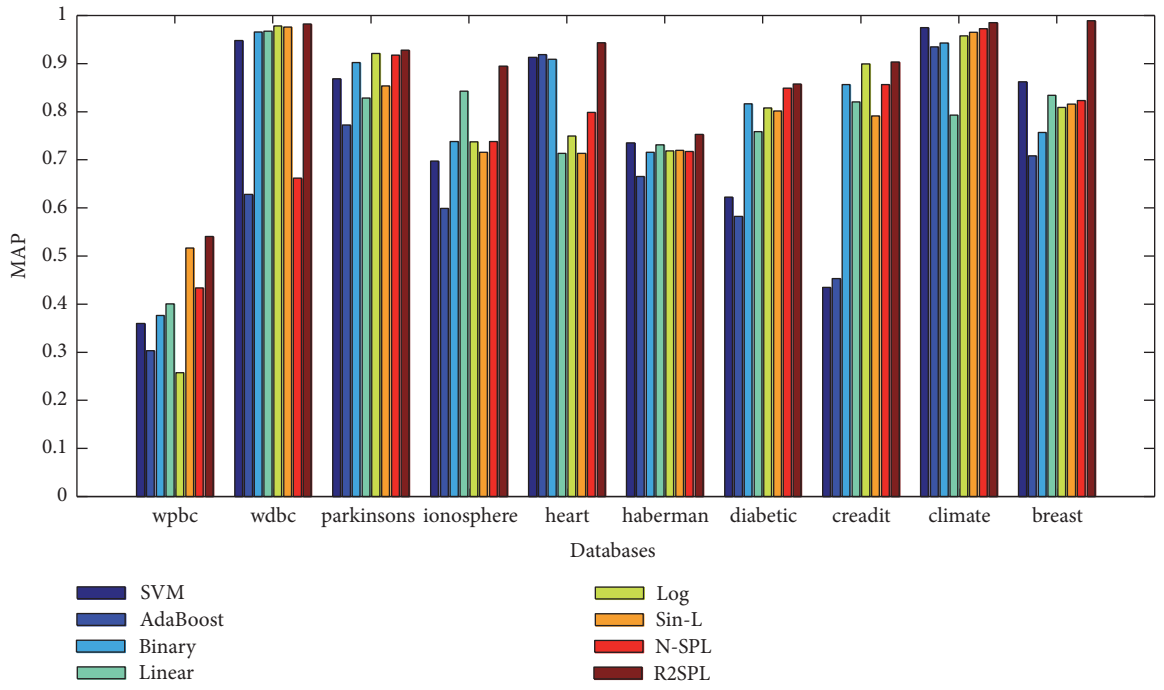


Figure 4: The classification results of state-of-art methods and our model on ten databases.

lists the ACC and AUC of seven baselines and our model. Our proposed model achieves all the best results on 10 UCI datasets and makes great improvement in several datasets.

We compared the proposed method, i.e., R2SPL, with several representative classification methods, including SVM, AdaBoost, SPL with binary, linear, or log function, Single-SPL (Sin-L), SPL without tough samples (N-SPL). The results are shown in Figure 4. We draw the precision-recall curves of these methods in Figure 5 and presented AUC and ACC results in Table 2. As seen from Figures 4 and 5 and Table 2, we find our methods outperform these comparison methods on all the ten datasets.

We also performed our method and comparison methods on ADNI dataset and conducted three classification tasks,

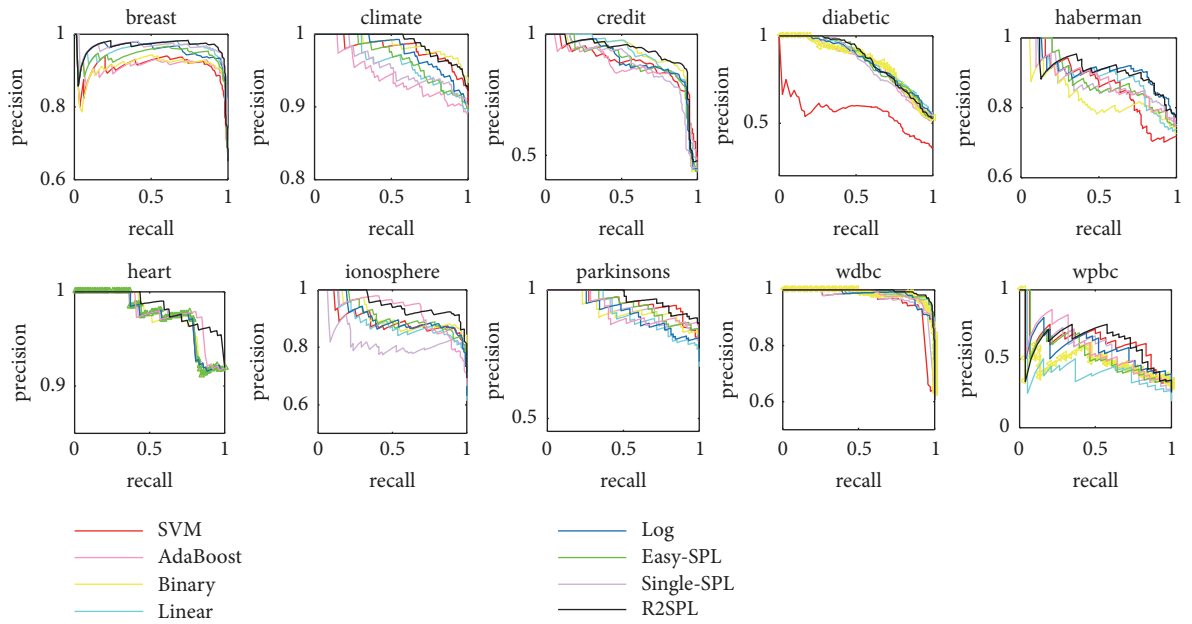


Figure 5: The precision-recall (PR) figures of state-of-the-art methods and our model on 10 UCI databases.

Table 3: Results of seven baselines and our model on the task of AD versus HC.

| | SVM | Ada-Boost | Binary | Linear | Log | N-SPL | Single-SPL | R2SPL |
|---|---|---|---|---|---|---|---|---|
| ACC | 78.31±0.034 | 87.45±0.021 | 74.12±0.019 | 71.80±0.028 | 71.29±0.056 | 89.57±0.033 | 90.09±0.057 | 91.34±0.071 |
| SEN | 89.09±0.028 | 90.83±0.047 | 75.56±0.035 | 66.38±0.044 | 79.79±0.068 | 90.57±0.027 | 90.46±0.026 | 92.38±0.059 |
| SPE | 62.21±0.019 | 82.90±0.051 | 73.88±0.080 | 80.32±0.037 | 60.94±0.038 | 89.64±0.068 | 90.62±0.087 | 90.99±0.096 |
| AUC | 80.70±0.022 | 83.74±0.034 | 74.65±0.029 | 72.78±0.011 | 70.12±0.042 | 90.35±0.019 | 91.68±0.027 | 92.62±0.028 |

Table 4: Results of seven baselines and our model on the task of MCI versus HC.

| | SVM | Ada-Boost | Binary | Linear | Log | N-SPL | Single-SPL | R2SPL |
|---|---|---|---|---|---|---|---|---|
| ACC | 66.61±0.021 | 83.68±0.028 | 71.77±0.059 | 73.52±0.027 | 73.01±0.019 | 80.32±0.097 | 80.12±0.075 | 84.09±0.069 |
| SEN | 50.24±0.035 | 80.24±0.034 | 48.27±0.022 | 62.12±0.038 | 56.84±0.057 | 70.13±0.032 | 71.62±0.062 | 73.42±0.095 |
| SPE | 82.99±0.068 | 81.56±0.019 | 76.35±0.021 | 84.91±0.044 | 89.18±0.051 | 90.52±0.058 | 88.61±0.078 | 94.76±0.063 |
| AUC | 67.16±0.026 | 82.49±0.038 | 57.40±0.051 | 70.68±0.035 | 71.21±0.080 | 78.96±0.049 | 82.58±0.072 | 86.49±0.082 |

Table 5: Results of seven baselines and our model on the task of SMC versus LMCI.

| | SVM | Ada-Boost | Binary | Linear | Log | N-SPL | Single-SPL | R2SPL |
|---|---|---|---|---|---|---|---|---|
| ACC | 69.50±0.043 | 66.89±0.042 | 73.63±0.028 | 72.08±0.053 | 75.07±0.034 | 69.53±0.053 | 69.86±0.066 | 75.81±0.033 |
| SEN | 46.75±0.029 | 48.63±0.036 | 54.70±0.055 | 59.46±0.032 | 65.16±0.056 | 48.00±0.035 | 60.94±0.087 | 48.32±0.016 |
| SPE | 70.13±0.093 | 73.59±0.028 | 74.18±0.090 | 69.03±0.046 | 70.71±0.042 | 72.87±0.024 | 66.48±0.055 | 80.72±0.097 |
| AUC | 71.40±0.045 | 64.74±0.014 | 74.79±0.038 | 74.53±0.031 | 79.40±0.089 | 64.93±0.048 | 69.48±0.019 | 72.93±0.029 |

AD versus HC, MCI versus HC, and SMC versus LMCI. The comparison results in three tasks, AD versus HC, MCI versus HC, and SMC versus LMCI, are listed in Tables 3, 4, and 5, respectively. Obviously, our proposed method has better performance compared with other methods in ACC, SEN, SPE, and AUC. It demonstrates the superiority of our model to other classifiers in AD classification problems.

*4.3. Parameter Influence.* Our model has two parameters including regularization term $\lambda$ and sparse term p. We test the influence of the two parameters on 10 UCI datasets. The parameter $\lambda$ is tuned from $10^{-3}$ to $10^{3}$ and the value of sparse term p is adjusted from 0 to 2. When detecting the sensitivity of a parameter to the model, we only change the value of this parameter and fix the value of another parameter. Figure 6 shows the experimental results. Specifically, Figure 6 shows the influence of regularization term $\lambda$ and parameter p. As we can see from Figure 6, when the $\lambda$ is tuned from $10^{-3}$ to $10^{3}$, the performance of our proposed model is stable in most cases. Figure 6 also shows the influence

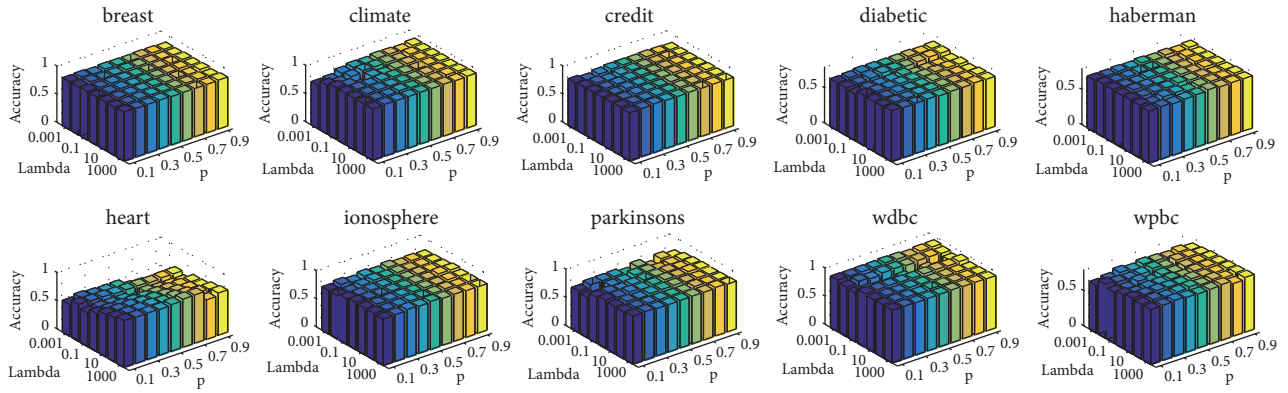


Figure 6: The results of our model base different parameters.

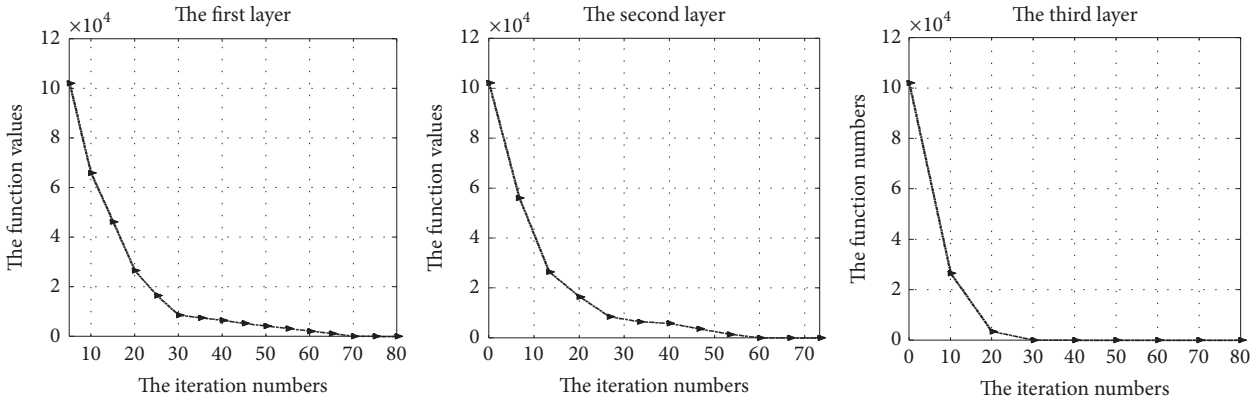


Figure 7: The convergence results of three layers models.

of sparse term p. When p changes from 0.4 to 2, the performance of our proposed method changes slightly. We conduct multiple groups of experiment on 10 UCI datasets. The experiment results verify that our model is not sensitive to specific parameters and only related to the structure of the model.

*4.4. The Convergence Results of Different Layers of Model.* In the convergence analysis, we find that different models have different rates of convergence. The convergence results are listed in Figure 7; obviously, as the number of layers increases, the convergence speed of the model is also accelerating. Benefiting from the prior knowledge of previous iteration process, the current model can obtain local optimal solution faster.

## 5. Conclusion

In this paper, we divide the samples into easy-learning samples, hard learning samples, and misclassified samples and analyze their roles in learning. Then, we introduce tough or misclassified sample in the training of each iteration to self-paced learning. Meanwhile, considering the human cognition process, people usually need to constantly explore and learn from the same data or task to obtain a deep knowledge about it by multiple learning stages. So, we design the retrospective framework to improve the robust of self-paced learning, which uses the model in previous layer to reduce the negative effect of small sample size problem in the initialization phase of next iteration. The experimental results show that the proposed method behaves more robust and discriminative than conventional self-paced learning methods and many representative methods. In our further work, we will extend above framework to other learning tasks, such as semisupervised learning.

## Data Availability

Raw data were generated at Nanjing University of Aeronautics and Astronautics. Derived data supporting the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by National Natural Science Foundation of China (nos. 61501230, 61732006,

61876082, and 61861130366), National Science and Technology Major Project (no. 2018ZX10201002), and the Fundamental Research Funds for the Central Universities (no. NP2018104).

## References

[1] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th International Conference On Machine Learning, ICML 2009*, pp. 41–48, Canada, June 2009.

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

[3] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann, "Self-paced curriculum learning," in *Proceedings of the AAAI*, vol. 2, p. 6, 2015.

[4] Q. Zhao, D. Meng, L. Jiang, Q. Xie, Z. Xu, and A. G. Hauptmann, "Self-paced learning for Matrix factorization," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI 2015 and the 27th Innovative Applications of Artificial Intelligence Conference, IAAI 2015*, pp. 3196–3202, 2015.

[5] J. S. Supancic III and D. Ramanan, "Self-paced learning for long-term tracking," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pp. 2379–2386, June 2013.

[6] C. Ying, M. Qi-Guang, L. Jia-Chen, and G. Lin, "Advance and prospects of AdaBoost algorithm," *Acta Automatica Sinica*, vol. 39, no. 6, pp. 745–758, 2013.

[7] X. Li, L. Wang, and E. Sung, "AdaBoost with SVM-based component classifiers," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 5, pp. 785–795, 2008.

[8] G. Tur, D. Hakkani-Tür, and R. E. Schapire, "Combining active and semi-supervised learning for spoken language understanding," *Speech Communication*, vol. 45, no. 2, pp. 171–186, 2005.

[9] S. Huang, R. Jin, and Z.-H. Zhou, "Active learning by querying informative and representative examples," *Advances in neural information processing systems*, pp. 892–900, 2010.

[10] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[11] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[12] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: algorithms and applications," *IEEE Access*, vol. 3, pp. 490–530, 2015.

[13] D.-C. Li, C.-S. Wu, T.-I. Tsai, and Y.-S. Lina, "Using mega-trend-diffusion and artificial samples in small data set learning for early flexible manufacturing system scheduling knowledge," *Computers & Operations Research*, vol. 34, no. 4, pp. 966–982, 2007.

[14] L. Jiang, D. Meng, T. Mitamura, and A. G. Hauptmann, "Easy samples first: Self-paced reranking for zero-example multimedia search," in *Proceedings of the 2014 ACM Conference on Multimedia, MM 2014*, pp. 547–556, USA, November 2014.

[15] S. Zhou, J. Wang, D. Meng et al., "Deep self-paced learning for person re-identification," *Pattern Recognition*, vol. 76, pp. 739–751, 2018.

[16] Y. Fan, R. He, J. Liang, and B.-G. Hu, "Self-paced learning: an implicit regularization perspective," in *Proceedings of the AAAI*, vol. 3, p. 4, 2017.

[17] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," *Statistics and Its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[18] F. Lv and R. Nevatia, "Recognition and segmentation of 3-d human action using hmm and multi-class adaboost," in *Proceedings of the European Conference on Computer Vision*, pp. 359–372, Springer, 2006.

[19] P. Viola and M. Jones, "Fast and robust classification using asymmetric adaboost and a detector cascade," *Advances in Neural Information Processing Systems*, pp. 1311–1318, 2002.

[20] G. Rätsch, T. Onoda, and K. R. Müller, "Soft margins for AdaBoost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, 2001.

[21] J. H. Morra, Z. Tu, L. G. Apostolova, A. E. Green, A. W. Toga, and P. M. Thompson, "Comparison of adaboost and support vector machines for detecting alzheimer's disease through automated hippocampal segmentation," *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 30–43, 2010.

[22] 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*, vol. 40, no. 1, pp. 7–19, 2018.

[23] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2001.

[24] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *Proceedings of the ICML*, pp. 839–846, Citeseer, 2000.

[25] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proceedings of the 9th ACM International Conference on Multimedia*, pp. 107–118, October 2001.

[26] P. Mitra, B. U. Shankar, and S. K. Pal, "Segmentation of multispectral remote sensing images using active support vector machines," *Pattern Recognition Letters*, vol. 25, no. 9, pp. 1067–1074, 2004.

[27] A. L. Brown and M. J. Kane, "Preschool children can learn to transfer: Learning to learn and learning from example," *Cognitive Psychology*, vol. 20, no. 4, pp. 493–523, 1988.

[28] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: a survey," *Journal of Machine Learning Research*, vol. 10, pp. 1633–1685, 2009.

[29] H. Shin, H. R. Roth, M. Gao et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[30] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 3156–3164, June 2015.