Big Data and Information Analytics
©American Institute of Mathematical Sciences
Volume 1, Number 1, January 2016

doi:10.3934/bdia.2016.1.111
pp. 111–127

WHY CURRICULUM LEARNING & SELF-PACED LEARNING WORK IN BIG/NOISY DATA: A THEORETICAL PERSPECTIVE

Tieliang Gong, Qian Zhao, Deyu Meng* and Zongben Xu

Institute for Information and System Sciences and Ministry of
Education Key Lab of Intelligent Networks and Network Security
Xi'an Jiaotong University
Xi'an, Shaanxi, China

Abstract. Since being recently raised, curriculum learning (CL) and self-paced learning (SPL) have attracted increasing attention due to its multiple successful applications. While currently the rationality of this learning regime is heuristically inspired by the cognitive principle of humans, there still isn't a sound theory to explain the intrinsic mechanism leading to its effectiveness, especially on some successful attempts on big/noise data. To address this issue, this paper presents some theoretical results for revealing the insights under this learning scheme. Specifically, we first formulate a new learning problem aiming to learn a proper classifier from samples generated from the training distribution which is deviated from the target distribution. Furthermore, we find that the CL/SPL regime provides a feasible solving strategy for this learning problem. Especially, by first introducing high-confidence/easy samples and gradually involving low-confidence/complex ones into learning, the CL/SPL process latently minimizes an upper bound of the expected risk under target distribution, purely using the data from the deviated training distribution. We further construct a new SPL learning algorithm based on random sampling, which better complies with our theory, and substantiate its effectiveness by experiments implemented on synthetic and real data.

1. Introduction.

Recently, *curriculum learning* (CL) [2] and *self-paced learning* (SPL) [12] have been attracting increasing attention in machine learning and computer vision. Both learning paradigms are inspired by the learning principle underlying the cognitive process of humans/animals, which generally starts with learning easier aspects of an learning task, and then gradually takes more complex examples into consideration.

Since being raised, multiple variations of this CL/SPL learning regime, like self-paced reranking [8], self-paced learning with diversity [9], and self-paced curriculum learning [10], have been proposed to further ameliorate its capability. Its effectiveness has also been extensively validated in various machine learning and computer vision tasks, including object detector adaptation [20], dictionary learning [19], long-term tracking [18] and matrix factorization [23]. Especially, this paradigm has

2010 *Mathematics Subject Classification.* Primary: 68Q32, 68T05; Secondary: 68T15.

Key words and phrases. Curriculum learning, self-paced learning, learning theory, TRECVID MED/MER, classification.

This research was supported by the National Grand Fundamental Research 973 Program of China under Grant No. 2013CB329404, the China NSFC project under grant 11571271, the China NSFC project under contract 61373114, 11131006.

* Corresponding author: Deyu Meng.

been integrated into the system developed by CMU Informedia team, and achieved the leading performance in challenging semantic query (SQ)/000Ex tasks of the TRECVID MED/MER competition organized by NIST in 2014 [22]. Just as indicated by the initial work [2] along this line, two advantages of the CL/SPL learning have been empirically substantiated, especially under big data/noisy scenarios [12, 8, 9, 10, 1, 11]: generalization improving and convergence speedup.

Albeit with superior performance in applications, the reasonability of the CL/SPL regime is only intuitively explained by its cognitive understanding, while short of a sound theory to reveal the insightful mechanism leading to its effectiveness. Specifically, current CL/SPL learning methods need to iteratively solve varying optimization problems under gradually increasing pace parameters [12, 8, 9, 10], while there is still not a theoretical argument presented to clarify where these methods converge to and which objective is these methods intrinsically solve.

To the above issue, this work initializes the learning theory for CL/SPL and provides an insightful explanation for the effectiveness mechanism under this line of learning schemes. Specifically, the main contribution of this paper can be summarized as the following aspects.

Different from the traditional learning theory assuming the similar training and test distribution, a new theory is formalized to understand the learning problem under the assumption that there exists deviation between training and test/target distributions. This actually is the case often encountered in this era of big data. Nowadays, in various learning tasks like object recognition, event detection and user behavior analysis, learners always need to achieve massive data source for training. In general these massive data are collected and annotated from company users (e.g., the Netflix database[1]), the web (e.g., the LFW database[2]) or by making use of crowdsourcing involvement (e.g., the ImageNet database[3]). The subjective understanding of any annotator is inevitably more-or-less deviated from the objective oracle knowledge underlying data. This naturally conducts the deviation from the training distribution (accumulated from knowledge of all involved annotators) and the true target one, especially in those ambiguous annotated regions. This inspires us to formulate this learning problem and investigate its learning theory.

Under the premise of the proposed learning theory, the insight of CL/SPL can be rationally explained. Especially, the theory clarifies that the CL/SPL regime actually attempts to minimize an upper bound of the expected risk under target distribution, purely from the data generated from the deviated training distribution. In specific, easy samples in CL/SPL correspond to those in high-confidence annotated area of training distribution, which is also consistent with the high-confidence region of the target distribution (where annotators can easily confirm and agree). Complex ones, however, are more likely to be located in the ambiguous annotated regions, corresponding to the more deviated area between training and target distributions (where users are easily get uncertain or even wrongly cognized). Thus to start training from easy samples by CL/SPL actually simulates learning from the high-confidence target region, while to gradually incrementing complex ones means that the samples residing on ambiguous training regions then come to be involved. Through this process, the faithful information delivered by those high-confidence/easy samples incline to soundly guide the learning towards the expected

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

[2] http://www.image-net.org/

[3] http://vis-www.cs.umass.edu/lfw/

target, while being less hampered by those low-confidence/complex samples relatively more deviated from the target. This naturally conducts the advantages of SPL, i.e., better generalization to target and faster convergence in a sound manner, as compared to the traditional learning mode, which considers or even emphasizes unreliable low-confidence samples throughout the learning process.

Besides, based on the proposed theory, we can construct a new CL/SPL learning scheme based on random sampling. This new scheme better complies with the deduced upper bound of the expected risk on the target distribution, and thus can be more faithfully explained by our theory. We also substantiate the effectiveness of the proposed learning scheme by experiments on synthetic dan real data.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work on CL/SPL. Section 3 introduces the new learning problem and our motivations. Section 4 establishes the main learning theory for this learning problem, and clarifies its intrinsic relationship to CL/SPL. The SPL learning algorithm by random sampling is constructed in Section 5, and evaluated by experiments in Section 6. The paper is then concluded with a future research.

2. **Related work.**

Inspired by the learning principle of humans/animals, [2] formulated the curriculum learning paradigm. Its core idea is to iteratively involve samples into learning in sequence, where easy samples are learned first and more complex ones are gradually included when the learner is ready for them. These gradually included sample sequences from easy to complex are called curriculums learned in different grown-up stages of training. In specific, [2] formalized the CL problem as follows. Let $P_{\text{train}}(\mathbf{z})$ be the training distribution from which the input data are generated, where $\mathbf{z}$ is a random variable representing a sample for the learner (corresponds to a pair of $(\mathbf{x}, y)$ for supervised learning). Let $0 \le W_\lambda(\mathbf{z}) \le 1$ be the weight superimposed on $\mathbf{z}$ at step λ in the curriculum sequence, with the pace parameter $0 \le \lambda \le 1$. The corresponding training distribution at step λ is

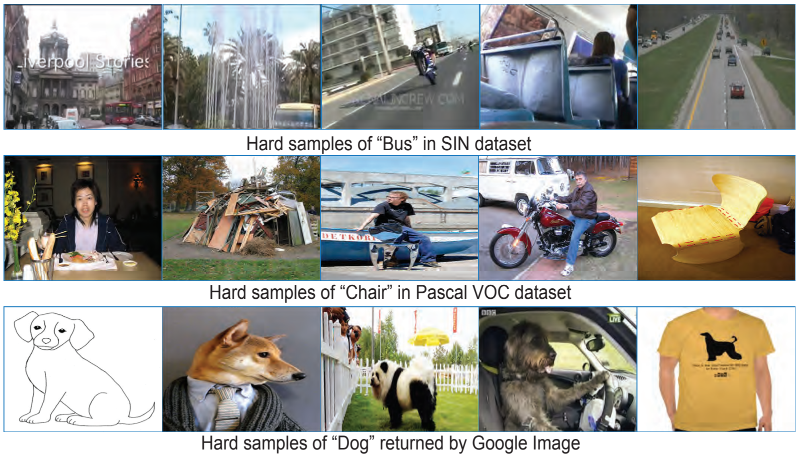
$$Q_\lambda(\mathbf{z}) \propto W_\lambda(\mathbf{z}) P_{\text{train}}(\mathbf{z}), \tag{1}$$

such that $\int_Z Q_\lambda(\mathbf{z}) d\mathbf{z} = 1$, where Z denotes the whole training set. A sequence $Q_\lambda(\mathbf{z})$ can be called a curriculum if it satisfies that both its entropy $H(Q_\lambda)$ and its weight function $W_\lambda(\mathbf{z})$ are monotonically increasing with respect to the increasing pace λ. This strategy has been empirically evaluated to be helpful in enhancing generalization capability and fastening the convergence speed in multiple applications [17, 1].

To make the CL idea more implementable in applications, [12] first formulated the key principle of CL as a concise optimization model named SPL. The SPL model includes a weighted loss term on all samples and a general SPL regularizer imposed on sample weights. By sequentially optimizing the model with gradually increasing pace parameter on the SPL regularizer, more samples can be automatically included into training from easy to complex in a pure self-paced way. [8] and [23] further built a guideline to construct a rational SPL regularizer, and formalized the SPL model as the following optimization problem:

$$\min_{\mathbf{w}, \mathbf{v} \in [0,1]^n} \sum_{i=1}^{n} v_i L(y_i, f(\mathbf{x}_i, \mathbf{w})) + r(\mathbf{v}; \lambda), \tag{2}$$

where $L(y, f(\mathbf{x}, \mathbf{w}))$ denotes the loss between the annotated label y and the estimated one $f(\mathbf{x}, \mathbf{w})$, with model parameter $\mathbf{w}$, and v_i denotes the binary variable,


Hard samples of “Bus” in SIN dataset

Hard samples of “Chair” in Pascal VOC dataset

Hard samples of “Dog” returned by Google Image

Figure 1: Some relatively complex samples from the SIN and Pascal VOC data sets, and returned by Google image search engine.

which indicates whether the i-th sample is easy or not. $r(\mathbf{v};\lambda)$ is the SPL regularizer. λ is a parameter controlling the learning pace. The larger λ is, the more samples are involved in training and the more “grown-up” the trained model is. Under this guide line, multiple variations of SPL models have been constructed, including self-paced reranking (SPaR) [8], self-paced learning with diversity [9], and self-paced curriculum learning [10], and multiple applications of this SPL framework have been attempted, such as object detector adaptation [20], specific-class segmentation learning [13], visual category discovery [14], long-term tracking [18] and background subtraction [23]. Especially, the SPaR method was integrated into the system developed by CMU Informedia team, and achieved leading performance in challenging SQ/000Ex tasks of the TRECVID MED/MER competition organized by NIST [22].

In this paper, we attempt to explore the insightful reason behind these successful applications of CL/SPL. To the best of our knowledge, this is the first theoretical explanation work for this newly emerging methodology.

3. A new understanding for the learning problem in big data sceneries.

The current learning tasks always need to collect a massive data set for training. Such a large magnitude makes it only possible to achieve the expected data from crowdsourcing, especially for supervised learning tasks. This often conducts large amount of ambiguous (or complex in CL/SPL) samples for general users in the obtained data, as illustrated in Figure 1, showing typical “hard” samples from the SIN[4] and Pascal VOC[5] data sets, and returned by Google image search engine[6]. The reason is that any participant has his/her own specific viewpoint on a problem as compared to most others, and there is thus inevitably a deviation from each collector/annotator’s subjective understanding to the objective oracle knowledge of

[4] http://www.ee.columbia.edu/ln/dvmm/a-TRECVID/

[5] http://host.robots.ox.ac.uk/pascal/VOC/

[6] https://images.google.com/

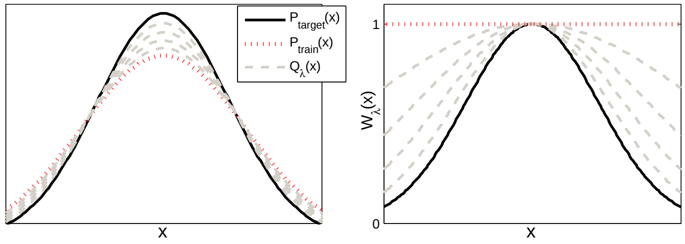


Figure 2: Left: Illustration for the training/target distribution $P_{\text{train}}(\mathbf{x})/P_{\text{target}}(\mathbf{x})$, as well as a sequence of pace distributions $Q_{\lambda}(\mathbf{x})$ varying from $P_{\text{target}}(\mathbf{x})$ to $P_{\text{train}}(\mathbf{x})$. Note that $P_{\text{train}}(\mathbf{x})$ has an evident heavy tail as compared to $P_{\text{target}}(\mathbf{x})$. Right: The corresponding weight functions with respect to varying pace λ.

the problem. This naturally leads to the problem that the training distribution, $P_{\text{train}}(\mathbf{z})$, accumulated by all collector/annotator's knowledge, is different from the test/target distribution, $P_{\text{target}}(\mathbf{z})$, to which the learning really needs to generalize.

Albeit deviated, useful information under $P_{\text{target}}(\mathbf{z})$ can still be explored from $P_{\text{train}}(\mathbf{z})$. Most participants share a same common sense on high-confidence samples, and these faithful samples thus tend to be distributed in a region with relatively large density. For supervised learning problem, such region should be located intra-class and relatively far from the classification boundary where samples are easy to be misclassified. In these high-confidence areas, the subjective understanding of humans and the objective knowledge should be consistent and $P_{\text{train}}(\mathbf{z})$ and $P_{\text{target}}(\mathbf{z})$ should be accordant. Comparatively, those ambiguous/complex samples, conducted by the cognitive differences or even misoperation of annotators, should occupy a relatively smaller proportion in data and located in a region with smaller density. Their locations should be near classification boundary or even inner wrong classes (e.g., noises/outliers) in supervised learning. This naturally leads to an evident heavy-tailed shape of $P_{\text{train}}(\mathbf{z})$ as compared to $P_{\text{target}}(\mathbf{z})$ in such low-confidence regions, as shown in Figure 2.

In small/clean sample cases, such a low-confidence region is always with few generated samples due to its small density and small base number of samples. Thus it tends to be configured as a blank "margin" area. Through finding a classification surface to maximize this margin, the decision boundary can always be effectively located [21]. In the premise of practical big/noisy data, however, such margin tends to be very hard to enanchor. Both relatively high density of marginal samples (caused by noise/outliers) and large data cardinality (caused by big data) tend to fill the margin, and the heavy noises/outliers even seriously mislead the margin location. This might explain the fail cases of traditional margin-emphasizing algorithms like SVM [21], Adaboost [7], and etc., in some real data applications [8, 9].

It is thus rational to more emphasize the high-confidence (i.e., easy) samples rather than low-confidence (i.e., complex) ones in certain real data cases, instead of treating the former as non-support-vectors and ignoring their role in learning. This constitutes the basic methodology under CL/SPL, which more complies with the human learning process. Such high-confidence-sample-emphasizing idea has also been employed to build never-ending machine learning systems that acquire the ability to extract structured information from unstructured data [4, 15] by persistently picking up high-confidence samples in iteration.

In sum, our argument is that in real big/noisy data scenarios, both learning theories and implementation methods need to be handled in new viewpoints. In theory, instead of similar [5, 6], the target distribution is often deviated from the training, especially in those low-confidence regions; and in implementation, high-confidence samples, i.e., the traditional non-support-vectors, might be put more emphasis in learning, as the CL/SPL methodology suggests.

In the following, we will provide some preliminary theoretical results on this new setting of learning problem, and deliver a rational theoretical explanation for the working mechanism under CL/SPL methodology.

4. **SPL learning theory.**

4.1. **Problem setting.** In this work we mainly investigate the binary classification problem. Following the classic setting of learning theory, our aimed learning problem is: Let X be a compact subset of $\mathbb{R}^d$, $Y = \{-1, 1\}$ be the label set and $Z = X \times Y$ be the whole set. The binary classification problem aims at learning a proper classifier $f : X \to Y$ from the input training samples $\{\mathbf{z}_i = (\mathbf{x}_i, y_i)\}_{i=1}^n$ generated from the underlying training distribution $P_{\text{train}}(Z) = P_{\text{train}}(X|Y)P_{\text{train}}(Y)$ [6], such that the following expected risk can be minimized:

$$\mathcal{R}(f) := \int_Z L_f(\mathbf{z})P_{\text{target}}(\mathbf{x}|y)P_{\text{target}}(y)d\mathbf{z},$$

where $P_{\text{target}}(Z) = P_{\text{target}}(X|Y)P_{\text{target}}(Y)$ denotes the target distribution on Z, and $L_f(\mathbf{z}) = \mathbb{1}_{f(\mathbf{x})\neq y} = \frac{1-yf(\mathbf{x})}{2}$, denoting the loss function measuring the difference between the predicted and true labels. Both $P_{\text{train}}(Z)$ and $P_{\text{target}}(Z)$ are fixed while unknown. The following empirical risk is thus considered for actual implementation:

$$\mathcal{R}_{emp}(f) = \frac{1}{n}\sum_{i=1}^n L_f(\mathbf{z}_i). \tag{3}$$

We assume $P_{\text{target}}(y = 1) = P_{\text{train}}(y = 1) = 1/2$ for easy evaluation and denote $P^+_{\text{train}}(\mathbf{x}) = P_{\text{train}}(\mathbf{x}|y = 1)$, $P^-_{\text{train}}(\mathbf{x}) = P_{\text{train}}(\mathbf{x}|y = -1)$, $P^+_{\text{target}}(\mathbf{x}) = P_{\text{target}}(\mathbf{x}|y = 1)$, $P^-_{\text{target}}(\mathbf{x}) = P_{\text{target}}(\mathbf{x}|y = -1)$. Since the deduction for both $y = 1$ and $y = -1$ cases are exactly similar, we only consider one case in the following and denote $P_{\text{train}}(\mathbf{x})$ and $P_{\text{target}}(\mathbf{x})$ omitting notion $+1$ or -1.

4.2. **A simulated curriculum format.** We first formulate $P_{\text{target}}(\mathbf{x})$ as the weighted expression of $P_{\text{train}}(\mathbf{x})$:

$$P_{\text{target}}(\mathbf{x}) = \frac{1}{\alpha^*}W_{\lambda^*}(\mathbf{x})P_{\text{train}}(\mathbf{x}), \tag{4}$$

where $0 \leq W_{\lambda^*}(\mathbf{x}) \leq 1$ and $\alpha^* = \int_X W_{\lambda^*}(\mathbf{x})P_{\text{train}}(\mathbf{x})d\mathbf{x}$ denotes the normalization factor[7]. Based on Eq. (4), $P_{\text{target}}(\mathbf{x})$ actually corresponds to a curriculum as defined in Eq. (1) under the weight function $W_{\lambda^*}(\mathbf{x})$. As analyzed in the last section, $W_{\lambda^*}(\mathbf{x})$ should be of small values in the low-confidence area of P_{target} where complex samples are located, while have larger values (close to 1) in the high-confidence area where easy samples reside. This can be easily understood by observing Figure 2.

Eq. (4) can be equivalently reformulated as

$$P_{\text{train}}(\mathbf{x}) = \alpha^* P_{\text{target}}(\mathbf{x}) + (1 - \alpha^*)E(\mathbf{x}) \tag{5}$$

[7] We thus have $\alpha^* \leq 1$ since $W_{\lambda^*}(\mathbf{x})P_{\text{train}}(\mathbf{x}) \leq P_{\text{train}}(\mathbf{x})$.

where

$$E(\mathbf{x}) = \frac{1}{1-\alpha^*}(1 - W_{\lambda^*}(\mathbf{x}))P_{\text{train}}(\mathbf{x}).$$

Here it is easy to see $E(\mathbf{x})$ is a distribution ($\int_X E(\mathbf{x})d\mathbf{x} = 1$) formulated by the weighted $P_{\text{train}}(\mathbf{x})$ under the weight function $(1 - W_{\lambda^*}(\mathbf{x}))$. This term actually measures the deviation from P_{target} to P_{train}. In high-confidence area of P_{target}, $E(\mathbf{x})$ corresponds to the nearly zero-weighted P_{train}, and thus the deviations/errors tend to be small. On the contrary, in the low-confidence area, $E(\mathbf{x})$ imposes relatively large weights on P_{train}, naturally leading to its large deviation values. This complies with our aforementioned analysis on the deviation measure. The more confidently a sample is annotated, the less deviated its label should be from the true one.

We can then construct the following curriculum sequence for our theoretical evaluation:

$$Q_\lambda(\mathbf{x}) = \alpha_\lambda P_{\text{target}}(\mathbf{x}) + (1-\alpha_\lambda)E(\mathbf{x}), \tag{6}$$

where α_λ varies from 1 to α^* with increasing pace parameter λ. Correspondingly, the curriculum Q_λ simulates the changing process from P_{target} to P_{train}, as illustrated in Figure 2. Note that $Q_\lambda(\mathbf{x})$ can also be regularized into the curriculum formulation as Eq. (1) as follows:

$$Q_\lambda(\mathbf{x}) \propto W_\lambda(\mathbf{x})P_{\text{train}}(\mathbf{x}),$$

where

$$W_\lambda(\mathbf{x}) \propto \frac{\alpha_\lambda P_{\text{target}}(\mathbf{x}) + (1-\alpha_\lambda)E(\mathbf{x})}{\alpha^* P_{\text{target}}(\mathbf{x}) + (1-\alpha^*)E(\mathbf{x})}$$

with $0 \leq W_\lambda(\mathbf{x}) \leq 1$ through normalizing its maximal value as 1.

Note that the initial stage of this CL process sets $W_\lambda \propto \frac{P_{\text{target}}}{P_{\text{train}}}$, which is of larger weights in the high-confidence area while much smaller in low-confidence area due to the heavy-tail problem. The weights are thus of more vibrations. With the pace λ increasing, the large weights in high-confidence area become smaller while small ones in low-confidence area become larger, leading to more uniform distributed weights with smaller variations. After normalizing $W_\lambda(\mathbf{x})$ into the interval $[0, 1]$, its values tend to consistently increase in λ, which can be easily understood by Figure 2. This thus complies with the weight-increasing condition defined for a curriculum in [2].

By taking (6) as the pace distribution, we attempt to present some theoretical results on CL/SPL strategy. These results will help us get some useful insights under this interesting learning scheme.

4.3. **CL/SPL learning theory.**

First we need some preliminary definitions.

Definition 4.1. Let $\mathcal{G}$ be a function family mapping from Z to $[a, b]$, $P(Z)$ a distribution on Z and $S = (\mathbf{z}_1, \cdots, \mathbf{z}_m)$ a set of i.i.d. samples drawn from P. The empirical Rademacher complexity of $\mathcal{G}$ with respect to S is then defined by

$$\hat{\mathfrak{R}}_m(\mathcal{G}) = \mathbb{E}_\sigma\big[\sup_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^m \sigma_i g(\mathbf{z}_i)\big], \tag{7}$$

where σ_is are i.i.d. samples drawn from the uniform distribution in $\{-1, 1\}$. The Rademacher complexity of $\mathcal{G}$ is defined by the expectation of $\hat{\mathfrak{R}}_m(\mathcal{G})$ over all samples S:

$$\mathfrak{R}_m(\mathcal{G}) = \mathbb{E}_{S\sim P^m}|\hat{\mathfrak{R}}_S(\mathcal{G})|. \tag{8}$$

Definition 4.2. The Kullback-Leibler divergence $D_{KL}(p\|q)$ between two densities $p(\Omega)$ and $q(\Omega)$ is defined by

$$D_{KL}(p\|q) = \int_{\Omega} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x}. \tag{9}$$

Based on the above definitions, we can estimate the generalization error bound for CL/SPL learning under the curriculum Q_λ. Firstly we present the following necessary lemmas for this task.

Lemma 4.3. *(Bretagnolle-Huber inequality) Let p and q be density functions, and then we have*

$$\int |p(\mathbf{x}) - q(\mathbf{x})| d\mathbf{x} \leq 2\sqrt{1 - \exp\{-D_{KL}(p \parallel q)\}}. \tag{10}$$

Lemma 4.4. [16] *Let $\mathcal{H}$ be a family of function taking value in $\{-1, 1\}$ and P be the distribution over the input space X. Then for any $\delta > 0$, with confidence at least $1 - \delta$ over a sample set S, the following holds for any $f \in \mathcal{H}$:*

$$\mathcal{R}(f) \leq \mathcal{R}_{emp}(f) + \mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\ln(1/\delta)}{2m}}. \tag{11}$$

In addition, we have

$$\mathcal{R}(f) \leq \mathcal{R}_{emp}(f) + \hat{\mathfrak{R}}_m(\mathcal{H}) + 3\sqrt{\frac{\ln(2/\delta)}{2m}}. \tag{12}$$

Lemma 4.5. *Suppose $S \subseteq \{\mathbf{x} : \|\mathbf{x}\| \leq R\}$ be a sample set of size m, and $\mathcal{H} = \{\mathbf{x} \longmapsto \text{sgn}(\mathbf{w}^T \cdot \mathbf{x}) : \min_S |\mathbf{w}^T x| = 1 \wedge \|\mathbf{w}\| \leq B\}$ be hypothesis class, where $\mathbf{w} \in \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^n$, and then we have*

$$\hat{\mathfrak{R}}_m(\mathcal{H}) \leq \frac{BR}{\sqrt{m}}. \tag{13}$$

Proof.

$$\begin{aligned}
\hat{\mathfrak{R}}_m(\mathcal{H}) &= \frac{1}{m}\mathbb{E}_\sigma\Big[\sup_{\|\mathbf{w}\|\leq B} \sum_{i=1}^m \sigma_i \text{sgn}(\mathbf{w}_i\mathbf{x}_i)\Big] \\
&\leq \frac{1}{m}\mathbb{E}_\sigma\Big[\sup_{\|\mathbf{w}\|\leq B} \sum_{i=1}^m \sigma_i |\text{sgn}(\mathbf{w}_i\mathbf{x}_i)|\Big] \\
&\leq \frac{1}{m}\mathbb{E}_\sigma\Big[\sup_{\|\mathbf{w}\|\leq B} \sum_{i=1}^m \sigma_i |\mathbf{w}_i\mathbf{x}_i|\Big] \leq \frac{B}{m}\mathbb{E}_\sigma\Big[\| \sum_{i=1}^m \sigma_i \mathbf{x}_i\|\Big] \\
&\leq \frac{B}{m}\mathbb{E}_\sigma\Big[[\| \sum_{i=1}^m \sigma_i \mathbf{x}_i\|^2]\Big]^{\frac{1}{2}} \\
&= \frac{B}{m}\mathbb{E}_\sigma\Big[[\| \sum_{i,j=1}^m \sigma_i\sigma_j(\mathbf{x}_i\mathbf{x}_j)\|^2]\Big]^{\frac{1}{2}} \\
&\leq \frac{B}{m}\Big[\mathbb{E}_\sigma[\| \sum_{i=1}^m \mathbf{x}_i\|^2]\Big]^{\frac{1}{2}} \\
&= \frac{BR}{\sqrt{m}}.
\end{aligned}$$

□

Then we give the main results of this work.

Theorem 4.6. *Suppose $\{\mathbf{z}_i\}_{i=1}^m$ are i.i.d. samples drawn from the pace distribution Q_λ. Let m_+/m_- be the number of positive/nagetive samples, $m^* = \min\{m_-, m_+\}$, and $\mathcal{H}$ the function family projecting to $\{-1, 1\}$. Then for any $\delta > 0$ and $f \in \mathcal{H}$, with confidence at least $1 - 2\delta$ we have:*

$$\begin{aligned}\mathcal{R}(f) &\leq \frac{1}{2}\mathcal{R}^+_{emp}(f) + \frac{1}{2}\mathcal{R}^-_{emp}(f)\\ &+ \frac{1}{2}\mathfrak{R}_{m_+}(\mathcal{H}) + \frac{1}{2}\mathfrak{R}_{m_-}(\mathcal{H}) + \sqrt{\frac{\ln(1/\delta)}{m^*}}\\ &+ (1-\alpha_\lambda)\sqrt{1 - \exp\{-D_{KL}(P^+_{\text{target}} \parallel E^+)\}}\\ &+ (1-\alpha_\lambda)\sqrt{1 - \exp\{-D_{KL}(P^-_{\text{target}} \parallel E^-)\}},\end{aligned} \tag{14}$$

and

$$\begin{aligned}\mathcal{R}(f) &\leq \frac{1}{2}\mathcal{R}^+_{emp}(f) + \frac{1}{2}\mathcal{R}^-_{emp}(f)\\ &+ \frac{1}{2}\hat{\mathfrak{R}}_{m_+}(\mathcal{H}) + \frac{1}{2}\hat{\mathfrak{R}}_{m_-}(\mathcal{H}) + 3\sqrt{\frac{\ln(2/\delta)}{m^*}} +\\ &+ (1-\alpha_\lambda)\sqrt{1 - \exp\{-D_{KL}(P^+_{\text{target}} \parallel E^+)\}}\\ &+ (1-\alpha_\lambda)\sqrt{1 - \exp\{-D_{KL}(P^-_{\text{target}} \parallel E^-)\}},\end{aligned} \tag{15}$$

where E^+, E^- denote the error distribution corresponding to P^+_{target}, P^-_{target}, and $\mathcal{R}^+_{emp}(f)$, $\mathcal{R}^-_{emp}(f)$ denote the empirical risk on positive samples and negative samples, respectively.

Proof. We first rewrite the expected risk as

$$\begin{aligned}\mathcal{R}(f) &= \int_Z L_f(\mathbf{z}) P_{target}(x|\mathbf{y}) P_{target}(\mathbf{y}) d\mathbf{z}\\ &= \frac{1}{2}\int_{X^+} L_f(\mathbf{x}, \mathbf{y}) P_{target}(\mathbf{x}|y = 1) d\mathbf{x}\\ &+ \frac{1}{2}\int_{X^-} L_f(\mathbf{x}, \mathbf{y}) P_{target}(\mathbf{x}|y = -1) d\mathbf{x}\\ &:= \frac{1}{2}(\mathcal{R}^+(f) + \mathcal{R}^-(f)).\end{aligned}$$

The empirical risk tends not to approximate the expected risk due to the inconsistence of P_{train} and P_{target}. However, by introducing intermediate risk with pace distribution, namely the pace risk, and denoting by $\mathbb{E}_{Q_\lambda}(f)$ in the error analysis, we can formulate the following error decomposition

$$\begin{aligned}&\frac{1}{2}(\mathcal{R}^+(f) + \mathcal{R}^-(f)) - \frac{1}{2}(\mathcal{R}^+_{emp}(f) + \mathcal{R}^-_{emp}(f))\\ &= \frac{1}{2}[\mathcal{R}^+(f) - \mathbb{E}_{Q^+_\lambda}(f) + \mathbb{E}_{Q^+_\lambda}(f) - \mathcal{R}^+_{emp}(f)]\\ &+ \frac{1}{2}[\mathcal{R}^-(f) - \mathbb{E}_{Q^-_\lambda}(f) + \mathbb{E}_{Q^-_\lambda}(f) - \mathcal{R}^-_{emp}(f)]\\ &:= S_1 + S_2.\end{aligned} \tag{16}$$

Let $S_1 = A_1 + A_2$ and $S_2 = B_1 + B_2$, where $A_1 = \frac{1}{2}(\mathcal{R}^+(f) - \mathbb{E}_{Q_\lambda^+}(f))$, $A_2 = \frac{1}{2}(\mathbb{E}_{Q_\lambda^+}(f) - \mathcal{R}^+_{emp})$, $B_1 = \frac{1}{2}(\mathcal{R}^-(f) - \mathbb{E}_{Q_\lambda^-}(f))$, $B_2 = \frac{1}{2}(\mathbb{E}_{Q_\lambda^-}(f) - \mathcal{R}^-_{emp}(f))$. Here, $\mathbb{E}_{Q_\lambda^+}(f)$ and $\mathbb{E}_{Q_\lambda^-}(f)$ denote the pace risk with respect to positive samples and negative samples, respectively.

We first focus on the estimation of A_1. By the fact the 0-1 loss is bounded by 1, we have

$$\begin{aligned}A_1 &\le \frac{1}{2}\int_{X_+}\big(P^+_{target}(\mathbf{x}) - Q^+_\lambda(\mathbf{x})\big)d\mathbf{x}\\ &= \frac{1}{2}\int_{X_+}\big(P^+_{target}(\mathbf{x}) - \alpha_\lambda P^+_{target}(\mathbf{x}) - (1-\alpha_\lambda)E^+(\mathbf{x})\big)d\mathbf{x}\\ &= \frac{1}{2}(1-\alpha_\lambda)\int_{X_+}\big(P^+_{target}(\mathbf{x}) - E^+(\mathbf{x})\big)d\mathbf{x}\\ &\le (1-\alpha_\lambda)\sqrt{1-\exp\{-D_{KL}(P^+_{target} \parallel E^+)\}}\end{aligned} \tag{17}$$

The last inequality is obtained by Lemma 4.3. For the estimation of A_2, according to Lemma 4.4, the following holds with confidence $1-\delta$

$$A_2 \le \frac{1}{2}\mathfrak{R}_{m_+}(\mathcal{H}) + \frac{1}{2}\sqrt{\frac{\ln(1/\delta)}{2m_+}}. \tag{18}$$

In the similar way, we can bound B_1 and B_2 as follows

$$B_1 \le (1-\alpha_\lambda)\sqrt{1-\exp\{-D_{KL}(P^-_{target} \parallel E^-)\}}, \tag{19}$$

and

$$B_2 \le \frac{1}{2}\mathfrak{R}_{m_-}(\mathcal{H}) + \frac{1}{2}\sqrt{\frac{\ln(1/\delta)}{2m_-}}. \tag{20}$$

By taking $m^* = \min\{m_+, m_-\}$ and combining Eqs. (17) (18) (19) (20), we can easily get Eq. (14). In addition, one can further get:

$$\mathfrak{R}_m(\mathcal{H}) \le \hat{\mathfrak{R}}_m(\mathcal{H}) + \sqrt{\frac{\ln(2/\delta)}{2m}}. \tag{21}$$

By replacing $\mathfrak{R}_m(\mathcal{H})$ in Eq. (14) with Eq. (21), we have (15).

The proof is then completed. □

Note that the above established error bounds upon 0-1 loss are hard to optimize. We thus further deduce another bound under the commonly utilized hinge loss.

Corollary 1. *Suppose $\{(\mathbf{x}_i, y_i)\}_{i=1}^m \subset (X \times \{-1,1\})$ are i.i.d. samples drawn from the pace distribution Q_λ with radius $|X| \le R$. Denote m_+/m_- be the number of positive/nagetive samples and $m^* = \min\{m_-, m_+\}$. Let $\mathcal{H} = \{x \longmapsto \mathbf{w}^T\mathbf{x} : \min_S |\mathbf{w}^T\mathbf{x}| = 1 \wedge \|\mathbf{w}\| \le B\}$, and $\phi(t) = (1-t)_+$ for $t \in \mathbb{R}$ be the hinge loss function. Then for any $\delta > 0$ and $g \in \mathcal{H}$, with confidence at least $1-2\delta$, it holds* *that:*

$$\begin{aligned}\mathcal{R}(\mathrm{sgn}(g)) \leq\ & \frac{1}{2m_+}\sum_{i=1}^{m_+}\phi(y_i g(\mathbf{x}_i)) + \frac{1}{2m_-}\sum_{i=1}^{m_-}\phi(y_i g(\mathbf{x}_i)) \\ & + \frac{RB}{\sqrt{m^*}} + 3\sqrt{\frac{\ln(1/\delta)}{m^*}} \\ & + (1-\alpha_\lambda)\sqrt{1-\exp\{-D_{KL}(P^+_{\text{target}} \parallel E^+)\}} \\ & + (1-\alpha_\lambda)\sqrt{1-\exp\{-D_{KL}(P^-_{\text{target}} \parallel E^-)\}}. \end{aligned} \tag{22}$$

Proof. Based on Lemma 4.5 to Eq. (15), and the fact that the hinge loss is the upper bound of $0-1$ loss, we can then obtain the result. □

Note that there are three components in the upper bound of the expected risk under P_{target}. The first row corresponds to the empirical risk on training samples generated from Q_λ. With λ increasing, these samples start by mainly generating from high-confident (easy) area of P_{target} in probability and gradually involve more complex ones. The second row reflects the approximation capability of training samples to evaluate information of Q_λ. The more samples are considered, the smaller this term is and the better approximation can be achieved. The last two rows measure the generalization capability of the learned classifier, which is monotonically increasing with respect to both the KL-divergence between the error distribution E and the target P_{target}, and the pace parameter λ. That is, the more deviated is the error E from P_{target}, the more difficult is to learn a proper classifier from training data which can generalize well on P_{target}. Also, in the late stage of CL/SPL (corresponding to large λ), the generalization of the learned classifier tends to be worse due to the gradually more evident deviation from the curriculum Q_λ to P_{target}. The last two terms actually compromise the approximation and generalization capabilities of this CL/SPL process with Q_λ.

This theory reveals the following insights underlying this CL/SPL process. The "easy-to-complex" property of the curriculum Q_λ intrinsically facilitates the information transfer from P_{train} to P_{target}, and makes it feasible to approximate the solution of the learning problem as set in Section 4.1, i.e., to learn a classifier with minimal expected risk on P_{target} through the empirical risk on training samples generated from P_{train}. In specific, we can approach the task of minimizing the expected risk on P_{target} by gradually increasing the pace λ, generating relatively high-confidence (easy) samples from Q_λ, and minimizing the empirical risk on these samples. This complies with the core idea under previous CL/SPL regimes. It is interesting that the previous investigations attribute the advantage of CL/SPL by that its performance is soundly guided by the faithful easy samples, while our theory further reveals that this regime facilitates learning to approach a good generalization to the target distribution.

5. **SPL insight: Approximate rational curriculums from training data.**

5.1. **Simulate Q_λ from training samples.**

When we only have samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n \subset X \times \{-1,1\}$ generated from P_{train}, we can approximately simulate a rational Q_λ as Eq. (6) in the following way. For easy discussion, we still only consider either of $+1$ and -1 cases, and ignore the notion $+1$ or -1.

First, let's approximate $\widehat{P}_{\text{train}} = p_i \delta_{\mathbf{x}_i}(\mathbf{x})$, where $\delta_{\mathbf{x}_i}(\mathbf{x})$ denotes the Dirac delta function centered at $\mathbf{x}_i$ and $p_i = \frac{1}{m}$. It is easy to see that $\widehat{P}_{\text{train}}$ supposes a uniform density on each sample $\mathbf{x}_i$. Next, in the beginning λ paces, we impose a smaller weights $v_i(\lambda)$ on low-confidence samples located near inter-class boundary than those on high-confidence regions to formulate the initial $\widehat{Q}_\lambda(\mathbf{x}) \propto \sum_{i=1}^n v_i(\lambda) p_i \delta_{\mathbf{x}_i}(\mathbf{x})$. By dominantly suppressing the heavy-tailed region of $\widehat{P}_{\text{train}}$, i.e., by putting nearly zero weights $v_i(\lambda)$ on those evident low-confidence samples, $\widehat{Q}_0$ is expected to form a rational approximation to P_{target}. We then increase the pace λ to gradually increase the small weight $v_i(\lambda)$ to 1. The corresponding $\widehat{Q}_\lambda(\mathbf{x}) \propto \sum_{i=1}^n v_i(\lambda) p_i \delta_{\mathbf{x}_i}(\mathbf{x})$ then approximates a curriculum sequence varying from $\widehat{Q}_0$ to $\widehat{P}_{\text{train}}$ like Eq. (6).

5.2. **Revisit previous SPL models.** Instead of minimizing the empirical risk $\mathcal{R}_{emp}(f)$ as illuminated in our theory, let's minimize its expected value under $\widehat{Q}_\lambda$ as:

$$
\begin{aligned}
&\min_{\mathbf{w}} \mathbb{E}_{\widehat{Q}_\lambda} \left(\frac{1}{n} \sum_{i=1}^n L(y_i, f(\mathbf{x}_i, \mathbf{w})) \right) = \mathbb{E}_{\widehat{Q}_\lambda} L(y, f(\mathbf{x}, \mathbf{w})) \\
&\Leftrightarrow \min_{\mathbf{w}} \sum_i v_i(\lambda) L(y_i, f(\mathbf{x}_i, \mathbf{w})),
\end{aligned}
\tag{23}
$$

where the first expectation is taken with respect to $\{\mathbf{x}_i\}_{i=1}^n$ which are i.i.d samples drawn from $\widehat{Q}_\lambda$. As analyzed above, $v_i(\lambda)$ should satisfy: (1) Under fixed λ, $v_i(\lambda)$ is monotonically increasing with its confidence degree; (2) For each sample $\mathbf{x}_i$, $v_i(\lambda)$ is monotonically increasing with respect to the pace λ.

An useful knowledge to judge whether the label confidence of a sample is high or low is through its learning error. That is, the high-confidence sample tends to be located inside the region of its category, thus always leading to its small training error, and vice versa. From this understanding, Eq. (23) exactly corresponds to current SPL learning models [8, 23, 10], which fit these weight values to accord with the similar requirements through supplementing a self-paced regularizer on $v_i(\lambda)$ in Eq. (23), as shown in the previous SPL model (2).

In this sense, we might explain the effectiveness of the previous SPL models by the following insight. Based on our theoretical results, this learning scheme tends to learn from the deviated training information to discover ground truth knowledge of the target distribution, through learning in a sound manner from high-confidence/easy/small-loss samples to low-confidence/complex/large-loss ones. Throughout this learning process, it intrinsically tries to minimize an upper bound of the expected risk on the target distribution, through being terminated at a proper compromised pace. This fully complies with the experience of its real implementations in multiple applications [8, 9, 23].

5.3. **SPL with random sampling.** Note that current SPL models are all deterministic, while the empirical risk in the upper bound (22) is calculated on randomly generated samples. We thus want to build a new SPL algorithm by using random sampling mechanism. The core idea is to approximate the pace distribution Q_λ by imposing weights on samples, and then sampling from this distribution to form new SPL training samples.

The implementation details are as follows. At each iteration, we first compute the losses of all training samples based on the current model. Then we solve the

Algorithm 1 Self-Pace Learning with Random Sampling (RS-SPL)

```
Input: training data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$, initial pace parameter $\lambda, m$ and stepsize $\mu, k$.
Output: model parameter $\mathbf{w}$.
Train a model on entire training set to obtain loss $\{L(y_i, f(\mathbf{x}_i, \mathbf{w}))\}_{i=1}^n$.
repeat
    Solve (24) to obtain $\mathbf{v}(\lambda)$.
    $\mathbf{v}(\lambda) = \mathbf{v}(\lambda)/\|\mathbf{v}(\lambda)\|_1$.
    Draw $m$ samples from $\sum_{i=1}^n v_i(\lambda) p_i \delta_{\mathbf{x}_i}(\mathbf{x})$ to form $\mathcal{D}_\lambda$.
    Train a new model on $\mathcal{D}_\lambda$ to obtain $\mathbf{w}$.
    If $\lambda$ is small, increase $\lambda$ by $\mu$ and increase $m$ by $k$.
until stopping criteria satisfied
```

following optimization problem to form weights on all samples:

$$\min_{\mathbf{v}} \sum_{i=1}^n v_i L(y_i, f(\mathbf{x}_i, \mathbf{w})) + r(\mathbf{v}, \lambda), \tag{24}$$

where $r(\mathbf{v}, \lambda)$ is the self-paced regularizer as defined in Eq. (2). After that, we normalize $\mathbf{v}$ by $\mathbf{v}/\|\mathbf{v}\|_1$ to construct the empirical pace distribution $\widehat{Q}_\lambda(\mathbf{x}) = v_i(\lambda) p_i \delta_{\mathbf{x}_i}(\mathbf{x})$, and then redraw samples from the training set according to $\widehat{Q}_\lambda$. A new model is then recursively trained on these samples. The whole process is summarized in Algorithm 1.

There are many choices for $r(\mathbf{v}, \lambda)$ based on three axiomic conditions defined on it [8]. We just readily use the following due to its easiness and effectiveness:

$$r(\mathbf{v}, \lambda) = -\gamma \sum_{i=1}^n \log\big(v_i + \frac{1}{\lambda}\gamma\big), \tag{25}$$

where $\gamma > 0$ is a tuning parameter. The optimal $\mathbf{v}(\lambda)$ to (24) can be analytically computed by

$$v_i(\lambda) = \begin{cases} \frac{1}{\log \gamma} \log(L(y_i, f(\mathbf{x}_i, \mathbf{w})) + \gamma) & L(y_i, f(\mathbf{x}_i, \mathbf{w})) < \lambda \\ 0 & L(y_i, f(\mathbf{x}_i, \mathbf{w})) \geq \lambda. \end{cases}$$

6. **Experiments.** In this section, we implemented experiments on synthetic and real classification datasets. The linear SVM, implemented by LibSVM [3], is utilized as the comparison method.

6.1. **A synthetic example.** We first give a synthetic example to illustrate behavior of the proposed RS-SPL algorithm. The data were generated as follows: Two 2-D Gaussian distributions, each associated with a class, were specified as the target distribution. The training distribution is further mixed with another two 2-D Gaussian distributions, each centered at the low density area of the target distribution of corresponding class to enforce deviation. We generated 2000 clean training samples, 1000 per class, and 2000 test samples from the target distributions. Then 400 samples from the deviated distributions, 200 per class, were added to the training set. The resulted training and test samples are shown in Figure 3.

In order to understand the behavior of RS-SPL, we implemented Algorithm 1 to this synthetic data and plot in Figure 4 the selected samples and the learned separating hyperplane during the SPL process. It can be observed that, samples

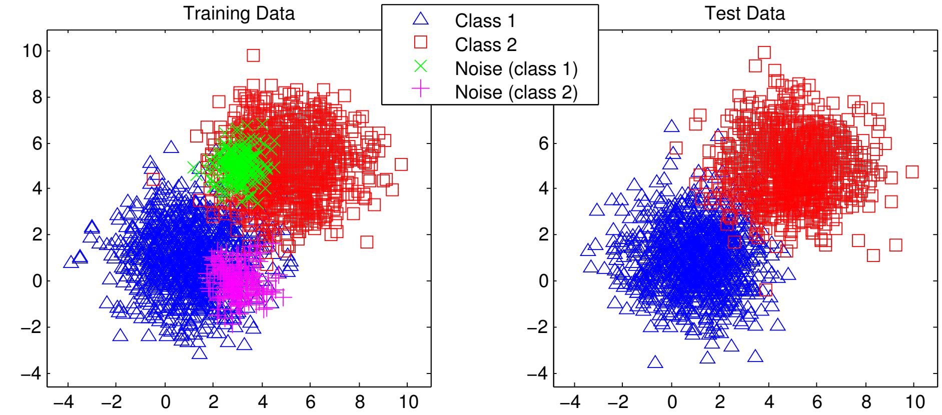


Figure 3: Samples used in our synthetic experiment. Left: Training samples. Triangles and squares are sampled from the target distribution, and crosses and pluses from the deviated distribution. Right: Test samples, generated from the target distribution.

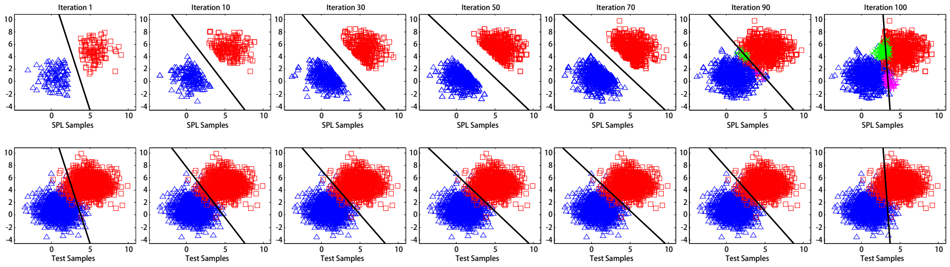


Figure 4: Upper: The selected training samples and the learned separating hyperplane (black line) in SPL iterations. Lower: Corresponding performance on the test samples.

from the high density region of the training distribution are selected first. As the SPL iteration continues, more and more samples with comparatively high confidence are included for training the classifier, and the separating hyperplane tends to be learned more accurately. However, when "hard" samples, i.e., those deviated samples, are included at the latter stages of SPL, the learned hyperplane tends to be disordered. Such behavior can also be substantiated by the accuracy tendency on the test data as shown in Figure 5. These results coincide with the SPL learning theory developed in Section 4, which asserts that the optimal expected risk tends to be achieved as a tradeoff between the better approximation capability of increasingly more samples and the worse generalization derived by the divergence from the pace distribution to the target.

6.2. **Real data evaluation.** We also implemented the proposed method to 5 real-world classification datasets, including *magic*[8], *image*, *waveform*, *ringnorm* and

[8] http://archive.ics.uci.edu/ml/datasets.html

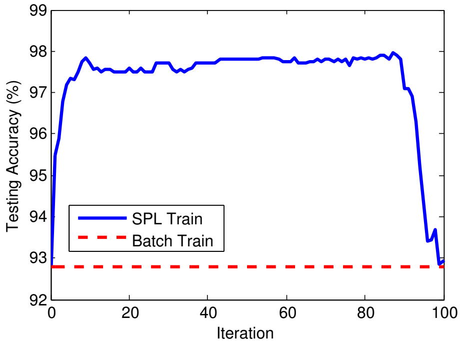


Figure 5: Classification accuracy (%) with respect to the SPL iteration in synthetic experiment.

Table 1: Statistics of 5 utilized real classification datasets.

Dataset	# Instances	# Features
magic	19020	10
waveform	5000	21
image	2310	18
ringnorm	7400	20
twonorm	7400	20

Table 2: Classification accuracy (%) on 5 real-world classification datasets. The results are averaged over 50 runs.

Dataset	# Batch Train	# SPL Train
magic	79.13 ± 0.28	79.74 ± 0.77
waveform	88.05 ± 0.52	88.30 ± 0.53
image	84.46 ± 1.11	86.26 ± 1.07
ringnorm	77.09 ± 0.63	77.36 ± 0.59
twonorm	97.71 ± 0.20	97.81 ± 0.17

twonorm[9]. The numbers of instances and features of each dataset are summarized in Table 1.

We randomly split each dataset into two subsets with equal sizes for training and testing, respectively. Then we applied the proposed RS-SPL algorithm to training a SVM classifier on the training set, and evaluated its performance in terms of classification accuracy on the test set. The parameters for SVM and RS-SPL were selected via hold-out validation on training set. We averaged the performance for each dataset over 50 runs as summarized in Table 2. As a comparison, we also include the results of the batch-trained SVM. We can see that the proposed SP-SPL algorithm can improve the classification accuracy over batch training. Its effectiveness can thus be validated.

[9] `http://www.raetschlab.org/Members/raetsch/benchmark`

7. **Conclusion.** We have presented a theoretical explanation for the working insight underlying the CL/SPL paradigm. Specifically, we clarify that the insight of the CL/SPL strategy is to learn knowledge of the target information from the given samples generated from the training distribution, which is deviated from the target. We have also argued that such a learning problem tends to happen in real big data scenarios due to the bias between subjective understanding of data collectors/annotators and objective oracle knowledge underlying data. Besides, our theory suggests the importance of high-confidence/easy samples in learning, which are generally taken as non-support-vectors in traditional learning methods and whose role is more or less underestimated. We further designed a new SPL algorithm with random sampling, which better complies our theory, and verified its effectiveness by experiments on synthetic and real data.

Our future research includes designing feasible termination condition for CL/SPL iteration based on our theory, deriving theory under unequal probabilities between $P(y = 1)$ and $P(y = -1)$, making the upper bound tighter, and applying the RS-SPL algorithm to more realistic big data sets.

REFERENCES

[1] S. Basu and J. Christensen, *Teaching Classification Boundaries to Humans,* Proceddings of the 27th AAAI Conference on Artificial Intelligence, 2013.

[2] Y. Bengio, J. Louradour, R. Collobert and J. Westone, Curriculum Learning, *Proceedings of the 26th International Conference on Machine Learning*, (2009), 41–48.

[3] C.-C. Chang and C.-J. Lin, LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology*, **2** (2011), 1–27. Software available from: `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] X. Chen, A. Shrivastava and A. Gupta, NEIL: Extracting visual knowledge from web data, *Proceedings of the IEEE International Conference on Computer Vision*, (2013), 1409–1416.

[5] F. Cucker and S. Smale, On the mathematical foundations of learning, *Bull. Amer. Math. Soc.*, **39** (2002), 1–49.

[6] F. Cucker and D. X. Zhou, *Learning Theory: An Approximation Theory Viewpoint*, Cambridge University Press, New York, NY, USA, 2007.

[7] Y. Freund and R. E. Schapire, Experiments with a new boosting algorithm, Proceedings of the 13th International Conference on Machine Learning, 1996.

[8] L. Jiang, D. Y. Meng, T. Mitamura and A. Hauptman, Easy samples first: Self-paced reranking for multimedia search, *Proceddings of the ACM International Conference on Multimedia*, (2014), 547–556.

[9] L. Jiang, D. Y. Meng, S. Yu, Z. Z. Lan, S. G. Shan and A. Hauptma, *Self-paced Learning with Diversity,* Advances in Nerual Information Processing Systems 27, 2014.

[10] L. Jiang and D. Y. Meng, Q. Zhao, S. G. Shan and A. Hauptman, *Self-paced Curriculum Learning,* Proceddings of the 29th AAAI Conference on Artificial Intelligence, 2015.

[11] F. Khan, X. Zhu and B. Mutlu, *How do Humans Teach: On Curriculum Learning and Teaching Dimension,* Advances in Nerual Information Processing Systems 24, 2011.

[12] M. Kumar, B. Packer and D. Koller, *Self-paced Learning for Latent Variable Models,* Advances in Nerual Information Processing Systems 23, 2010.

[13] M. Kumar, H. Turki, D. Preston and D. Koller, Learning specfic-class segmentation from diverse data, Proceedings of the IEEE International Conference on Computer Vision, 2011.

[14] Y. Lee and K. Grauman, Learning the easy things first: Self-paced visual category discovery, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2011), 1721–1728.

[15] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves and J. Welling, *Never-Ending Learning,* Proceddings of the 29th AAAI Conference on Artificial Intelligence, 2015.

[16] M. Mohri, A. Rostamizadeh and A. Talwalkar, *Foundations of Machine Learning*, The MIT Press, Cambridge, Massachusetts, London, England, 2012.

[17] E. Ni and C Ling, Supervised learning with minimal effort, *Advances in Knowledge Discovery and Data Mining*, **6119** (2010), 476–487.

[18] J. Supanvcivc and D. Ramana, Self-paced learning for long-term tracking, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[19] Y. Tang, Y. B. Yang and Y. Gao, Self-paced Dictionary Learning for Image Classification, *Proceddings of the ACM International Conference on Multimedia*, (2012), 833–836.

[20] K. Tang, V. Ramanathan, F. Li and D. Koller, Shifting weights: Adapting object detectors from image to video, *Advances in Nerual Information Processing Systems 25*, 2012.

[21] V. Vapnik, *Statistical Learning Theory*, Wiley-Interscience, New York, 1998.

[22] S. Yu, L. Jiang, Z. Mao, X. J. Chang, X. Z. Du, C. Gan, Z. Z. Lan, Z. W. Xu, X. C. Li, Y. Cai, A. Kumar, Y. Miao, L. Martin, N. Wolfe, S. C. Xu, H. Li, M. Lin, Z. G. Ma, Y. Yang, D. Y. Meng, S. G. Shan, P. D. Sahin, S. Burger, F. Metze, R. Singh, B. Raj, T. Mitamura, R. Stern and A. Hauptmann, *CMU-Informedia@ TRECVID 2014 Multimedia Event* Detection (MED), TRECVID Video Retrieval Evaluation Workshop, 2014.

[23] Q. Zhao, D. Y. Meng, L. Jiang, Q. Xie, Z. B. Xu and A. Hauptman, *Self-paced Matrix Factorization,* Proceddings of the 29th AAAI Conference on Artificial Intelligence, 2015.

Received May 2015; revised August 2015.

E-mail address: adidasgtl@gmail.com
E-mail address: timmy.zhaoqian@gmail.com
E-mail address: dymeng@mail.xjtu.edu.cn
E-mail address: zbxu@mail.xjtu.edu.cn