Self-Paced Curriculum Learning

Lu Jiang[1], Deyu Meng[1,2], Qian Zhao[1,2], Shiguang Shan[1,3], Alexander G. Hauptmann[1]
[1] School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 15217
[2] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, Shaanxi, P. R. China, 710049
[3] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, P. R. China, 100190
lujiang@cs.cmu.edu, dymeng@mail.xjtu.edu.cn,
timmy.zhaoqian@gmail.com, sgshan@ict.ac.cn, alex@cs.cmu.edu

Abstract

Curriculum learning (CL) or self-paced learning (SPL) represents a recently proposed learning regime inspired by the learning process of humans and animals that gradually proceeds from easy to more complex samples in training. The two methods share a similar conceptual learning paradigm, but differ in specific learning schemes. In CL, the curriculum is predetermined by prior knowledge, and remain fixed thereafter. Therefore, this type of method heavily relies on the quality of prior knowledge while ignoring feedback about the learner. In SPL, the curriculum is dynamically determined to adjust to the learning pace of the leaner. However, SPL is unable to deal with prior knowledge, rendering it prone to overfitting. In this paper, we discover the missing link between CL and SPL, and propose a unified framework named self-paced curriculum leaning (SPCL). SPCL is formulated as a concise optimization problem that takes into account both prior knowledge known before training and the learning progress during training. In comparison to human education, SPCL is analogous to "instructor-student-collaborative" learning mode, as opposed to "instructor-driven" in CL or "student-driven" in SPL. Empirically, we show that the advantage of SPCL on two tasks.

Curriculum learning (Bengio et al. 2009) and *self-paced learning* (Kumar, Packer, and Koller 2010) have been attracting increasing attention in the field of machine learning and artificial intelligence. Both the learning paradigms are inspired by the learning principle underlying the cognitive process of humans and animals, which generally start with learning easier aspects of a task, and then gradually take more complex examples into consideration. The intuition can be explained in analogous to human education in which a pupil is supposed to understand elementary algebra before he or she can learn more advanced algebra topics. This learning paradigm has been empirically demonstrated to be instrumental in avoiding bad local minima and in achieving a better generalization result (Khan, Zhu, and Mutlu 2011; Basu and Christensen 2013; Tang et al. 2012).

A curriculum determines a sequence of training samples which essentially corresponds to a list of samples ranked in ascending order of learning difficulty. A major disparity between *curriculum learning* (CL) and *self-paced learning* (SPL) lies in the derivation of the curriculum. In CL, the curriculum is assumed to be given by an oracle beforehand, and remains fixed thereafter. In SPL, the curriculum is dynamically generated by the learner itself, according to what the learner has already learned.

The advantage of CL includes the flexibility to incorporate prior knowledge from various sources. Its drawback stems from the fact that the curriculum design is determined independently of the subsequent learning, which may result in inconsistency between the fixed curriculum and the dynamically learned models. From the optimization perspective, since the learning proceeds iteratively, there is no guarantee that the predetermined curriculum can even lead to a converged solution. SPL, on the other hand, formulates the learning problem as a concise biconvex problem, where the curriculum design is embedded and jointly learned with model parameters. Therefore, the learned model is consistent. However, SPL is limited in incorporating prior knowledge into learning, rendering it prone to overfitting. Ignoring prior knowledge is less reasonable when reliable prior information is available. Since both methods have their advantages, it is difficult to judge which one is better in practice.

In this paper, we discover the missing link between CL and SPL. We formally propose a unified framework called *Self-paced Curriculum Leaning* (SPCL). SPCL represents a general learning paradigm that combines the merits from both the CL and SPL. On one hand, it inherits and further generalizes the theory of SPL. On the other hand, SPCL addresses the drawback of SPL by introducing a flexible way to incorporate prior knowledge. This paper also discusses concrete implementations within the proposed framework, which can be useful for solving various problems.

This paper offers a compelling insight on the relationship between the existing CL and SPL methods. Their relation can be intuitively explained in the context of human education, in which SPCL represents an "instructor-student collaborative" learning paradigm, as opposed to "instructor-driven" in CL or "student-driven" in SPL. In SPCL, instructors provide prior knowledge on a weak learning sequence of samples, while leaving students the freedom to decide the actual curriculum according to their learning pace. Since an optimal curriculum for the instructor may not necessarily be optimal for all students, we hypothesize that given reasonable prior knowledge, the curriculum devised by instructors and students together can be expected to be better than the curriculum designed by either part alone. Empirically, we substantiate this hypothesis by demonstrating that the proposed method outperforms both CL and SPL on two tasks.

The rest of the paper is organized as follows. We first briefly introduce the background knowledge on CL and SPL. Then we propose the model and the algorithm of SPCL. After that, we discuss concrete implementations of SPCL. The experimental results and conclusions are presented in the last two sections.

Background Knowledge

Curriculum Learning

Bengio et al. proposed a new learning paradigm called *curriculum learning* (CL), in which a model is learned by gradually including from easy to complex samples in training so as to increase the entropy of training samples (Bengio et al. 2009). Afterwards, Bengio and his colleagues presented insightful explorations for the rationality underlying this learning paradigm, and discussed the relationship between CL and conventional optimization techniques, e.g., the continuation and annealing methods (Bengio, Courville, and Vincent 2013; Bengio 2014). From human behavioral perspective, evidence have shown that CL is consistent with the principle in human teaching (Khan, Zhu, and Mutlu 2011; Basu and Christensen 2013).

The CL methodology has been applied to various applications, the key in which is to find a ranking function that assigns learning priorities to training samples. Given a training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i$ denotes the i^{th} observed sample, and y_i represents its label. A curriculum is characterized by a ranking function γ. A sample with a higher rank, i.e., smaller value, is supposed to be learned earlier.

The curriculum (or the ranking function) is often derived by predetermined heuristics for particular problems. For example, in the task of classifying geometrical shapes, the ranking function was derived by the variability in shape (Bengio et al. 2009). The shapes exhibiting less variability are supposed to be learned earlier. In (Khan, Zhu, and Mutlu 2011), the authors tried to teach a robot the concept of "graspability" - whether an object can be grasped and picked up with one hand, in which participants were asked to assign a learning sequence of graspability to various object. The ranking is determined by common sense of the participants. In (Spitkovsky, Alshawi, and Jurafsky 2009), the authors approached grammar induction, where the ranking function is derived in terms of the length of a sentence. The heuristic is that the number of possible solutions grows exponentially with the length of the sentence, and short sentences are easier and thus should be learn earlier.

The heuristics in these problems turn out to be beneficial. However, the heuristical curriculum design may lead to inconsistency between the fixed curriculum and the dynamically learned models. That is, the curriculum is predetermined a priori and cannot be adjusted accordingly, taking into account the feedback about the learner.

Self-paced Learning

To alleviate the issue of CL, Koller's group (Kumar, Packer, and Koller 2010) designed a new formulation, called *self-paced learning* (SPL). SPL embeds curriculum design as a regularization term into the learning objective. Compared with CL, SPL exhibits two advantages: first, it jointly optimizes the learning objective together with the curriculum, and therefore the curriculum and the learned model are consistent under the same optimization problem; second, the regularization term is independent of loss functions of specific problems. This theory has been successfully applied to various applications, such as action/event detection (Jiang et al. 2014b), reranking (Jiang et al. 2014a), domain adaption (Tang et al. 2012), dictionary learning (Tang, Yang, and Gao 2012), tracking (Supančič III and Ramanan 2013) and segmentation (Kumar et al. 2011).

Formally, let $L(y_i, g(\mathbf{x}_i, \mathbf{w}))$ denote the loss function which calculates the cost between the ground truth label y_i and the estimated label $g(\mathbf{x}_i, \mathbf{w})$. Here $\mathbf{w}$ represents the model parameter inside the decision function g. In SPL, the goal is to jointly learn the model parameter $\mathbf{w}$ and the latent weight variable $\mathbf{v} = [v_1, \cdots, v_n]^T$ by minimizing:

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

where λ is a parameter for controlling the learning pace. Eq. (1) indicates the loss of a sample is discounted by a weight. The objective of SPL is to minimize the weighted training loss together with the negative l_1-norm regularizer $-\|\mathbf{v}\|_1 = -\sum_{i=1}^n v_i$ (since $v_i \geq 0$). A more general regularizer consists of both $\|\mathbf{v}\|_1$ and $\|\mathbf{v}\|_{2,1}$ (Jiang et al. 2014b).

ACS (Alternative Convex Search) is generally used to solve Eq. (1) (Gorski, Pfeuffer, and Klamroth 2007). It is an iterative method for biconvex optimization, in which the variables are divided into two disjoint blocks. In each iteration, a block of variables are optimized while keeping the other block fixed. With the fixed $\mathbf{w}$, the global optimum $\mathbf{v}^* = [v_1^*, \cdots, v_n^*]$ can be easily calculated by:

$$v_i^* = \begin{cases} 1, & L(y_i, g(\mathbf{x}_i, \mathbf{w})) < \lambda, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

There exists an intuitive explanation behind this alternative search strategy: first, when updating $\mathbf{v}$ with a fixed $\mathbf{w}$, a sample whose loss is smaller than a certain threshold λ is taken as an "easy" sample, and will be selected in training ($v_i^* = 1$), or otherwise unselected ($v_i^* = 0$); second, when updating $\mathbf{w}$ with a fixed $\mathbf{v}$, the classifier is trained only on the selected "easy" samples. The parameter λ controls the pace at which the model learns new samples, and physically λ corresponds to the "age" of the model. When λ is small, only "easy" samples with small losses will be considered. As λ grows, more samples with larger losses will be gradually appended to train a more "mature" model.

This strategy complies with the heuristics in most CL methods (Bengio et al. 2009; Khan, Zhu, and Mutlu 2011). However, since the learning is completely dominated by the training loss, the learning may be prone to overfitting. Moreover, it provides no way to incorporate prior guidance in learning. To the best of our knowledge, there has been no studies to incorporate prior knowledge into SPL, nor to analyze the relation between CL and SPL.

Self-paced Curriculum Learning

Model and Algorithm

An ideal learning paradigm should consider both prior knowledge known before training and information learned during training in a unified and sound framework. Similar to human education, we are interested in constructing an "instructor-student collaborative" paradigm, which, on one hand, utilizes prior knowledge provided by instructors as a guidance for curriculum design (the underlying CL methodology), and, on the other hand, leaves students certain freedom to adjust to the actual curriculum according to their learning paces (the underlying SPL methodology).

This requirement can be realized through the following optimization model. Similar in CL, we assume that the model is given a curriculum that is predetermined by an oracle. Following the notation defined above, we have:

$$\min_{\mathbf{w},\mathbf{v}\in[0,1]^n} \mathbb{E}(\mathbf{w},\mathbf{v};\lambda,\Psi) = \sum_{i=1}^{n} v_i L(y_i, g(\mathbf{x}_i,\mathbf{w})) + f(\mathbf{v};\lambda) \quad \text{s.t. } \mathbf{v}\in\Psi \tag{3}$$

where $\mathbf{v} = [v_1, v_2, \cdots, v_n]^T$ denote the weight variables reflecting the samples' importance. f is called self-paced function which controls the learning scheme; Ψ is a feasible region that encodes the information of a predetermined curriculum. A curriculum can be mathematically described as:

Definition 1 (Total order curriculum) *For training samples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, a total order curriculum, or curriculum for short, can be expressed as a ranking function:*

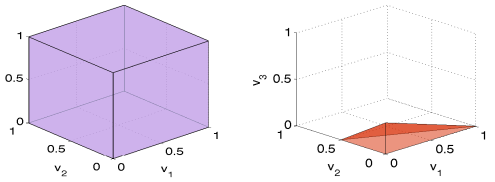
$$\gamma : \mathbf{X} \to \{1, 2, \cdots, n\},$$

where $\gamma(\mathbf{x}_i) < \gamma(\mathbf{x}_j)$ represents that x_i should be learned earlier than x_j in training. $\gamma(\mathbf{x}_i) = \gamma(\mathbf{x}_j)$ denotes there is no preferred learning order on the two samples.

Definition 2 (Curriculum region) *Given a predetermined curriculum $\gamma(\cdot)$ on training samples $\mathbf{X} = \{x_i\}_{i=1}^n$ and their weight variables $\mathbf{v} = [v_1, \cdots, v_n]^T$. A feasible region Ψ is called a curriculum region of γ if*

1. *Ψ is a nonempty convex set;*
2. *for any pair of samples $\mathbf{x}_i, \mathbf{x}_j$, if $\gamma(\mathbf{x}_i) < \gamma(\mathbf{x}_j)$, it holds that $\int_\Psi v_i \, d\mathbf{v} > \int_\Psi v_j \, d\mathbf{v}$, where $\int_\Psi v_i \, d\mathbf{v}$ calculates the expectation of v_i within Ψ. Similarly if $\gamma(\mathbf{x}_i) = \gamma(\mathbf{x}_j)$, $\int_\Psi v_i \, d\mathbf{v} = \int_\Psi v_j \, d\mathbf{v}$.*

The two conditions in Definition 2 offer a realization for curriculum learning. Condition 1 ensures the soundness for calculating the constraints. Condition 2 indicates that samples to be learned earlier should have larger expected values. The curriculum region physically corresponds to a convex region in the high-dimensional space. The area inside this region confines the space for learning the weight variables. The shape of the region weakly implies a prior learning sequence of samples, where the expected values for favored samples are larger. For example, Figure 1(b) illustrates an example of feasible region in 3D where the x, y, z axis represents the weight variable v_1, v_2, v_3, respectively. Without considering the learning objective, we can see that v_1 tends to be learned earlier than v_2 and v_3. This is because if we uniformly sample sufficient points in the feasible region of the coordinate (v_1, v_2, v_3), the expected value of v_1 is larger. Since prior knowledge is missing in Eq. (1), the feasible region is a unit hypercube, i.e. all samples are equally favored, as shown in Figure 1(a). Note the curriculum region should be confined within the unit hypercube since the constraints $\mathbf{v} \in [0,1]^n$ in Eq. (3).



(a) SPL (b) SPCL

Figure 1: Comparison of feasible regions in SPL and SPCL.

Note that the prior learning sequence in the curriculum region only weakly affects the actual learning sequence, and it is very likely that the prior sequence will be adjusted by the learners. This is because the prior knowledge determines a weak ordering of samples that suggests what should be learned first. A learner takes this knowledge into account, but has his/her own freedom to alter the sequence in order to adjust to the learning objective. See an example in the supplementary materials. Therefore, SPCL represents an "instructor-student-corporative" learning paradigm.

Compared with Eq. (1), SPCL generalizes SPL by introducing a regularization term. This term determines the learning scheme, i.e., the strategy used by the model to learn new samples. In human learning, we tend to use different schemes for different tasks. Similarly, SPCL should also be able to utilize different learning schemes for different problems. Since the existing methods only include a single learning scheme, we generalize the learning scheme and define:

Definition 3 (Self-paced function) *A self-paced function determines a learning scheme. Suppose that $\mathbf{v} = [v_1, \cdots, v_n]^T$ denotes a vector of weight variable for each training sample and $\ell = [\ell_1, \cdots, \ell_n]^T$ are the corresponding loss. λ controls the learning pace (or model "age"). $f(\mathbf{v};\lambda)$ is called a self-paced function, if*

1. *$f(\mathbf{v};\lambda)$ is convex with respect to $\mathbf{v} \in [0,1]^n$.*
2. *When all variables are fixed except for v_i, ℓ_i, v_i^* decreases with ℓ_i, and it holds that $\lim_{\ell_i \to 0} v_i^* = 1$, $\lim_{\ell_i \to \infty} v_i^* = 0$.*
3. *$\|\mathbf{v}\|_1 = \sum_{i=1}^n v_i$ increases with respect to λ, and it holds that $\forall i \in [1, n]$, $\lim_{\lambda \to 0} v_i^* = 0$, $\lim_{\lambda \to \infty} v_i^* = 1$.*

where $\mathbf{v}^* = \arg\min_{\mathbf{v}\in[0,1]^n} \sum v_i \ell_i + f(\mathbf{v};\lambda)$, and denote $\mathbf{v}^* = [v_1^*, \cdots, v_n^*]$.

The three conditions in Definition 3 provide a definition for the self-paced learning scheme. Condition 2 indicates that the model inclines to select easy samples (with smaller

Table 1: Comparison of different learning approaches.

	CL	SPL	Proposed SPCL
Comparable to human learning	Instructor-driven	Student-driven	Instructor-student collaborative
Curriculum design	Prior knowledge	Learning objective	Learning objective + prior knowledge
Learning schemes	Multiple	Single	Multiple
Iterative training	Heuristic approach	Gradient-based	Gradient-based

losses) in favor of complex samples (with larger losses). Condition 3 states that when the model "age" λ gets larger, it should incorporate more, probably complex, samples to train a "mature" model. The convexity in Condition 1 ensures the model can find good solutions within the curriculum region.

It is easy to verify that the regularization term in Eq. (1) satisfies Definition 3. In fact, this term corresponds to a binary learning scheme since v_i can only take binary values, as shown in the closed-form solution of Eq. (2). This scheme may be less appropriate in the problems where the importance of samples needs to be discriminated. In fact, there exist a plethora of self-paced functions corresponding to various learning schemes. We will detail some of them in the next section.

Inspired by the algorithm in (Kumar, Packer, and Koller 2010), we propose a similar ACS algorithm to solve Eq. (3). Algorithm 1 takes the input of a predetermined curriculum, an instantiated self-paced function and a stepsize parameter; it outputs an optimal model parameter $\mathbf{w}$. First of all, it represents the input curriculum as a curriculum region that follows Definition 2, and initializes variables in their feasible region. Then it alternates between two steps until it finally converges: Step 4 learns the optimal model parameter with the fixed and most recent $\mathbf{v}^*$; Step 5 learns the optimal weight variables with the fixed $\mathbf{w}^*$. In first several iterations, the model "age" is increased so that more complex samples will be gradually incorporated in the training. For example, we can increase λ so that μ more samples will be added in the next iteration. According to the conditions in Definition 3, the number of complex samples increases along with the growth of the number iteration. Step 4 can be conveniently implemented by existing off-the-shelf supervised learning methods. Gradient-based or interior-point methods can be used to solve the convex optimization problem in Step 5. According to (Gorski, Pfeuffer, and Klamroth 2007), the alternative search in Algorithm 1 converges as the objective function is monotonically decreasing and is bounded from below.

Relationship to CL and SPL

SPCL represents a general learning framework which includes CL and SPL as special cases. SPCL degenerates to SPL when the curriculum region is ignored ($\Psi = [0,1]^n$), or equivalently, the prior knowledge on predefined curriculums is absent. In this case, the learning is totally driven by the learner. SPCL degenerates to CL when the curriculum region (feasible region) only contains the learning sequence in the predetermined curriculum. In this case, the learning process neglects the feedback about learners, and is dominated by the given prior knowledge. When information from both sources are available, the learning in SPCL is collaboratively driven by prior knowledge and learning objective. Table 1 summarizes the characteristics of different learning methods. Given reasonable prior knowledge, SPCL which considers the information from both sources tend to yield better solutions. The toy example in supplementary materials lists a case in this regard.

Algorithm 1: Self-paced Curriculum Learning.

```
input  : Input dataset $\mathcal{D}$, predetermined curriculum
         $\gamma$, self-paced function $f$ and a stepsize $\mu$
output: Model parameter $\mathbf{w}$
1 Derive the curriculum region $\Psi$ from $\gamma$;
2 Initialize $\mathbf{v}^*$, $\lambda$ in the curriculum region;
3 while not converged do
4     Update $\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathbb{E}(\mathbf{w}, \mathbf{v}^*; \lambda, \Psi)$;
5     Update $\mathbf{v}^* = \arg\min_{\mathbf{v}} \mathbb{E}(\mathbf{w}^*, \mathbf{v}; \lambda, \Psi)$;
6     if $\lambda$ is small then increase $\lambda$ by the stepsize $\mu$;
7     ;
8 end
9 return $\mathbf{w}^*$
```

SPCL Implementation

The definition and algorithm in the previous section provide a theoretical foundation for SPCL. However, we still need concrete self-paced functions and curriculum regions to solve specific problems. To this end, this section discusses some implementations that follow Definition 2 and Definition 3. Note that there is no single implementation that can always work the best for all problems. As a pilot work on this topic, our purpose is to argument the implementations in the literature, and to help enlighten others to further explore this interesting direction.

Curriculum region implementation: We suggest an implementation induced from a linear constraint for realizing the curriculum region: $\mathbf{a}^T\mathbf{v} \leq c$, where $\mathbf{v} = [v_1, \cdots, v_n]^T$ are the weight variables in Eq. (3), c is a constant, and $\mathbf{a} = [a_1, \cdots, a_n]^T$ is a n-dimensional vector. The linear constraints is a simple implementation for curriculum region that can be conveniently solved. It can be proved that this implementation complies with the definition of curriculum region. See the proof in supplementary materials.

Theorem 1 *For training samples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, given a curriculum γ defined on it, the feasible region, defined by,*

$$\Psi = \{\mathbf{v} | \mathbf{a}^T\mathbf{v} \leq c\}$$

is a curriculum region of γ if it holds: 1) $\Psi \wedge \mathbf{v} \in [0,1]^n$ is nonempty; 2) $a_i < a_j$ for all $\gamma(\mathbf{x}_i) < \gamma(\mathbf{x}_j)$; $a_i = a_j$ for all $\gamma(\mathbf{x}_i) = \gamma(\mathbf{x}_j)$.

Self-paced function implementation: Similar to the scheme human used to absorb knowledge, a self-paced function determines a learning scheme for the model to learn new samples. Note the self-paced function is realized as a regularization term, which is independent of specific loss functions, and can be easily applied to various problems. Since human tends to use different learning schemes for different tasks, SPCL should also be able to utilize different learning schemes for different problems. Inspired by a study in (Jiang et al. 2014a), this section discusses some examples of learning schemes.

Binary scheme: This scheme in is used in (Kumar, Packer, and Koller 2010). It is called binary scheme, or "hard" scheme, as it only yields binary weight variables.

$$f(\mathbf{v};\lambda) = -\lambda \|v\|_1 = -\lambda \sum_{i=1}^{n} v_i, \tag{4}$$

Linear scheme: A common approach is to linearly discriminate samples with respect to their losses. This can be realized by the following self-paced function:

$$f(\mathbf{v};\lambda) = \frac{1}{2}\lambda \sum_{i=1}^{n} (v_i^2 - 2v_i), \tag{5}$$

in which $\lambda > 0$. This scheme represents a "soft" scheme as the weight variable can take real values.

Logarithmic scheme: A more conservative approach is to penalize the loss logarithmically, which can be achieved by the following function:

$$f(\mathbf{v};\lambda) = \sum_{i=1}^{n} \zeta v_i - \frac{\zeta^{v_i}}{\log \zeta}, \tag{6}$$

where $\zeta = 1 - \lambda$ and $0 < \lambda < 1$.

Mixture scheme: Mixture scheme is a hybrid of the "soft" and the "hard" scheme (Jiang et al. 2014a). If the loss is either too small or too large, the "hard" scheme is applied. Otherwise, the soft scheme is applied. Compared with the "soft" scheme, the mixture scheme tolerates small errors up to a certain point. To define this starting point, an additional parameter is introduced, i.e. $\lambda = [\lambda_1, \lambda_2]^T$. Formally,

$$f(\mathbf{v};\lambda) = -\zeta \sum_{i=1}^{n} \log(v_i + \frac{1}{\lambda_1}\zeta), \tag{7}$$

where $\zeta = \frac{\lambda_1 \lambda_2}{\lambda_1 - \lambda_2}$ and $\lambda_1 > \lambda_2 > 0$.

Theorem 2 *The binary, linear, logarithmic and mixture scheme function are self-paced functions.*

It can be proved that the above functions follow Definition 3. The name of the learning scheme suggests the characteristic of its solution. For example, denote $\ell_i = L(y_i, g(\mathbf{x}_i, \mathbf{w}))$. When $\Psi = [0,1]^n$, the partial gradient of Eq. (3) using logarithmic scheme equals:

$$\frac{\partial \mathbb{E}_{\mathbf{w}}}{\partial v_i} = \ell_i + (\zeta - \zeta^{v_i}) = 0, \tag{8}$$

where $\mathbb{E}_{\mathbf{w}}$ denote the objective in Eq. (3) with the fixed $\mathbf{w}$. We then can easily deduce:

$$\log(\ell_i + \zeta) = v_i \log \zeta. \tag{9}$$

The optimal solution for $\mathbb{E}_{\mathbf{w}}$ is given by:

$$v_i^* = \begin{cases} \frac{1}{\log \zeta} \log(\ell_i + \zeta) & \ell_i < \lambda \\ 0 & \ell_i \geq \lambda. \end{cases} \tag{10}$$

As shown the solution of v_i^* is logarithmic to its loss ℓ_i. See supplementary materials for the analysis on other self-paced functions. When the curriculum region is not a unit hypercube, the closed-form solution, such as Eq. (10), cannot be directly used. Gradient-based methods can be applied. As $\mathbb{E}_{\mathbf{w}}$ is convex, the local optimal is also the global optimal solution for the subproblem.

Experiments

We present experimental results for the proposed SPCL on two tasks: matrix factorization and multimedia event detection. We demonstrate that our approach outperforms baseline methods on both tasks.

Matrix Factorization

Matrix factorization (MF) aims to factorize an $m \times n$ data matrix $\mathbf{Y}$, whose entries are denoted as y_{ij}s, into two smaller factors $\mathbf{U} \in \mathbb{R}^{m \times r}$ and $\mathbf{V} \in \mathbb{R}^{n \times r}$, where $r \ll \min(m, n)$, such that $\mathbf{U}\mathbf{V}^T$ is possibly close to $\mathbf{Y}$ (Chatzis 2014; Meng et al. 2013; Zhao et al. 2014). MF has many successful applications, such as structure from motion (Tomasi and Kanade 1992) and photometric stereo (Hayakawa 1994). Here we test SPCL scheme on synthetic MF problems.

The data were generated as follows: two matrices $\mathbf{U}$ and $\mathbf{V}$, both of which are of size 40×4, were first randomly generated with each entry drawn from the Gaussian distribution $\mathcal{N}(0,1)$, leading to a ground truth rank-4 matrix $\mathbf{Y}_0 = \mathbf{U}\mathbf{V}^T$, and certain amount of noises were then specified to constitute the observation matrix $\mathbf{Y}$. Specifically, 20% of the entries were added to uniform noise on $[-50, 50]$, other 20% were added to uniform noise on $[-40, 40]$, and the rest were added to Gaussian noise drawn from $\mathcal{N}(0, 0.1^2)$. We considered L_2- and L_1-norm MF methods, and incorporated the SPL and SPCL frameworks with the solvers proposed by Cabral et al. (2013) and Wang et al. (2012), respectively. The curriculum region was constructed by setting the weight vector $\mathbf{v}$ and c in the linear constraint as follows. For $\mathbf{v}$, first, set $\tilde{v}_{ij} = 50$ for entries mixed with uniform noise on $[-50, 50]$, $\tilde{v}_{ij} = 40$ for entries mixed with uniform noise on $[-40, 40]$, and $\tilde{v}_{ij} = 1$ for the rest. Then $\mathbf{v}$ was calculated by $v_{ij} = \frac{\tilde{v}_{ij}}{\sum \tilde{v}_{ij}}$. For c, we specified it as 0.02 and 0.01 for L_2 and L_1-norm MF, respectively.

Two criteria were used for performance assessment. (1) *root mean square error* (RMSE): $\frac{1}{\sqrt{mn}} \|\mathbf{Y}_0 - \hat{\mathbf{U}}\hat{\mathbf{V}}^T\|_F$, and (2) *mean absolute error* (MAE): $\frac{1}{mn} \|\mathbf{Y}_0 - \hat{\mathbf{U}}\hat{\mathbf{V}}^T\|_1$, where $\hat{\mathbf{U}}, \hat{\mathbf{V}}$ denote the output from a utilized MF method. The performance of each method was evaluated as the average over 50 random realizations, as summarized in Table 2.

Table 2: Performance comparison of SPCL and baseline methods for matrix factorization.

	L_2-norm MF			L_1-norm MF		
	Baseline	SPL	SPCL	Baseline	SPL	SPCL
RMSE	9.3908	0.2585	**0.0654**	2.8671	0.1117	**0.0798**
MAE	6.8597	0.0947	**0.0497**	1.4729	0.0766	**0.0607**

The results show that the baseline methods fail to obtain reasonable approximation to the ground truth matrices due to the large noises embedded in the data, while SPL and SPCL significantly improve the performance. Besides, SPCL outperforms SPL. This is because SPL is more sensitive to the starting values than SPCL, and inclines to overfit to the noises. In this case, SPCL can alleviate such issue, as depicted in Figure 2. Because SPCL is constrained by prior curriculum and can weight the noisy samples properly.

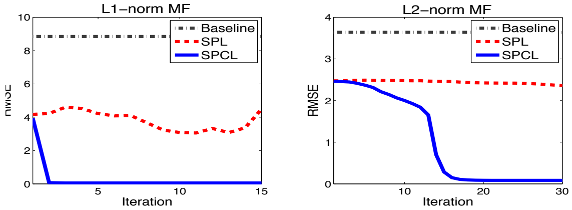


Figure 2: Comparison of the convergence of SPL and SPCL.

Multimedia Event Detection (MED)

Given a collection of videos, the goal of MED is to detect events of interest, e.g. "Birthday Party" and "Parade", solely based on the video content. Since MED is a very challenging task, there have been many studies proposed to tackle this problem in different settings, which includes training detectors using sufficient examples (Wang et al. 2013; Gkalelis and Mezaris 2014; Tong et al. 2014), using only a few examples (Safadi, Sahuguet, and Huet 2014; Jiang et al. 2014b), by exploiting semantic features (Tan, Jiang, and Neo 2014; Liu et al. 2013; Zhang et al. 2014; Inoue and Shinoda 2014; Jiang, Hauptmann, and Xiang 2012; Tang et al. 2012; Yu, Jiang, and Hauptmann 2014; Cao et al. 2013), and by automatic speech recognition (Miao, Metze, and Rawat 2013; Miao et al. 2014; Chiu and Rudnicky 2013).

We applied SPCL in a reranking setting, in which zero examples are given. It aims at improving the ranking of the initial search result. TRECVID Multimedia Event Detection (MED) 2013 Development, MED13Test and MED14Test sets were used (Over et al. 2013), which include around 34,000 Internet videos. The performance was evaluated on the MED13Test and MED14Test sets (25,000 videos), by the Mean Average Precision (MAP). There were 20 prespecified events on each dataset. Six types of visual and acoustic features were used. More information about these features is in (Jiang et al. 2014c).

In CL, the curriculum was derived by the MMPRF (Jiang et al. 2014c). In SPL, the curriculum was derived by the learning objective according to Eq. (1) where the loss is the hinge loss. In SPCL, Algorithm 1 was used, where Step 5 was solved by LM-BFGS (Zhu et al. 1997) in "stats" package in the R language, and Step 4 was solved by a standard quadratic programming toolkit. Mixture scheme was used, and all parameters were carefully tuned on a validation set on a different set of events. The predetermined curriculum in MMPRF was encoded as linear constraints $\mathbf{A}\mathbf{v} \leq \mathbf{g}$ to encode prior knowledge on modality weighting presented in (Jiang et al. 2014c). The intuition is that some features are more discriminative than others, and the constraints emphasize these discriminative features.

As we see in Table 3, SPCL outperforms both CL and SPL. The improvement is statistically significant across 20 events at the p-level of 0.05, according to the paired t-test. For this problem, "student-driven" learning mode (SPL) turns out better than "instructor-driven" mode (CL). "Instructor-student-collaborative" learning mode exploits prior knowledge and improves SPL. We hypothesize the reason is that SPCL takes advantage of the reliable prior knowledge and thus arrives at better solutions. The results substantiate the argument that learning with both prior knowledge and learning objective tends to be beneficial.

Table 3: Performance comparison of SPCL and baseline methods for zero-example event reranking.

Dataset	CL	SPL	SPCL
MED13Test	10.1	10.8	**12.9**
MED14Test	7.3	8.6	**9.2**

Conclusions and Future Work

We proposed a novel learning regime called self-paced curriculum learning (SPCL), which imitates the learning regime of humans/animals that gradually involves from easy to more complex training samples into the learning process. The proposed SPCL can exploit both prior knowledge before training and dynamical information extracted during training. The novel regime is analogous to an "instructor-student-collaborative" learning mode, as opposed to "instructor-driven" in curriculum learning or "student-driven" in self-paced learning. We presented compelling understandings for curriculum learning and self-paced learning, and revealed that they can be unified into a concise optimization model. We discussed several concrete implementations in the proposed SPCL framework. Experimental results on two different tasks substantiate the advantage of SPCL. Empirically, we found that SPCL requires a validation set that follows the same underlying distribution of the test set for tuning parameters in some problems. Intuitively, the set is analogous to the mock exam in education whose purposes are to let students realize how well they would perform on the real test, and, importantly, have a better idea of what to study.

Future directions may include developing new learning schemes for different problems. Since human tends to use different learning schemes to solve different problems, SPCL should utilize appropriate learning schemes for various problems at hand. Besides, currently as in curriculum learning, we assume the curriculum is total-order. We plan to relax this assumption in our future work.

Acknowledgments

This paper was partially supported by the US Department of Defense, U. S. Army Research Office (W911NF-13-1-0277) and by the National Science Foundation under Grant No. IIS-1251187. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ARO, the National Science Foundation or the U.S. Government.

References

Basu, S., and Christensen, J. 2013. Teaching classification boundaries to humans. In *AAAI*.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *ICML*.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on PAMI* 35(8):1798–1828.

Bengio, Y. 2014. Evolving culture versus local minima. In *Growing Adaptive Machines*. Springer. 109–138.

Cabral, R.; De la Torre, F.; Costeira, J. P.; and Bernardino, A. 2013. Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition. In *ICCV*.

Cao, L.; Gong, L.; Kender, J. R.; Codella, N. C.; and Smith, J. R. 2013. Learning by focusing: A new framework for concept recognition and feature selection. In *ICME*.

Chatzis, S. P. 2014. Dynamic bayesian probabilistic matrix factorization. In *AAAI*.

Chiu, J., and Rudnicky, A. 2013. Using conversational word bursts in spoken term detection. In *Interspeech*.

Gkalelis, N., and Mezaris, V. 2014. Video event detection using generalized subclass discriminant analysis and linear support vector machines. In *ICMR*.

Gorski, J.; Pfeuffer, F.; and Klamroth, K. 2007. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research* 66(3):373–407.

Hayakawa, H. 1994. Photometric stereo under a light source with arbitrary motion. *Journal of the Optical Society of America A* 11(11):3079–3089.

Inoue, N., and Shinoda, K. 2014. n-gram models for video semantic indexing. In *MM*.

Jiang, L.; Meng, D.; Mitamura, T.; and Hauptmann, A. G. 2014a. Easy samples first: Self-paced reranking for zero-example multimedia search. In *MM*.

Jiang, L.; Meng, D.; Yu, S.-I.; Lan, Z.; Shan, S.; and Hauptmann, A. G. 2014b. Self-paced learning with diversity. In *NIPS*.

Jiang, L.; Mitamura, T.; Yu, S.-I.; and Hauptmann, A. G. 2014c. Zero-example event search using multimodal pseudo relevance feedback. In *ICMR*.

Jiang, L.; Hauptmann, A. G.; and Xiang, G. 2012. Leveraging high-level and low-level features for multimedia event detection. In *MM*.

Khan, F.; Zhu, X.; and Mutlu, B. 2011. How do humans teach: On curriculum learning and teaching dimension. In *NIPS*.

Kumar, M.; Turki, H.; Preston, D.; and Koller, D. 2011. Learning specific-class segmentation from diverse data. In *ICCV*.

Kumar, M.; Packer, B.; and Koller, D. 2010. Self-paced learning for latent variable models. In *NIPS*.

Liu, J.; Yu, Q.; Javed, O.; Ali, S.; Tamrakar, A.; Divakaran, A.; Cheng, H.; and Sawhney, H. 2013. Video event recognition using concept attributes. In *WACV*.

Meng, D.; Xu, Z.; Zhang, L.; and Zhao, J. 2013. A cyclic weighted median method for l_1 low-rank matrix factorization with missing entries. In *AAAI*.

Miao, Y.; Jiang, L.; Zhang, H.; and Metze, F. 2014. Improvements to speaker adaptive training of deep neural networks. In *SLT*.

Miao, Y.; Metze, F.; and Rawat, S. 2013. Deep maxout networks for low-resource speech recognition. In *ASRU*.

Over, P.; Awad, G.; Michel, M.; Fiscus, J.; Sanders, G.; Kraaij, W.; Smeaton, A. F.; and Quenot, G. 2013. TRECVID 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID*.

Safadi, B.; Sahuguet, M.; and Huet, B. 2014. When textual and visual information join forces for multimedia retrieval. In *ICMR*.

Spitkovsky, V. I.; Alshawi, H.; and Jurafsky, D. 2009. Baby steps: How less is more in unsupervised dependency parsing. In *NIPS*.

Supančič III, J., and Ramanan, D. 2013. Self-paced learning for long-term tracking. In *CVPR*.

Tan, S.; Jiang, Y.-G.; and Neo, C.-W. 2014. Placing videos on a semantic hierarchy for search result navigation. *TOMCCAP* 10(4).

Tang, K.; Ramanathan, V.; Li, F.; and Koller, D. 2012. Shifting weights: Adapting object detectors from image to video. In *NIPS*.

Tang, Y.; Yang, Y. B.; and Gao, Y. 2012. Self-paced dictionary learning for image classification. In *MM*.

Tomasi, C., and Kanade, T. 1992. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision* 9(2):137–154.

Tong, W.; Yang, Y.; Jiang, L.; Yu, S.-I.; Lan, Z.; Ma, Z.; Sze, W.; Younessian, E.; and Hauptmann, A. G. 2014. E-lamp: integration of innovative ideas for multimedia event detection. *Machine vision and applications* 25(1):5–15.

Wang, N.; Yao, T.; Wang, J.; and Yeung, D. 2012. A probabilistic approach to robust matrix factorization. In *ECCV*.

Wang, F.; Sun, Z.; Jiang, Y.; and Ngo, C. 2013. Video event detection using motion relativity and feature selection. *IEEE Transactions on Multimedia*.

Yu, S.-I.; Jiang, L.; and Hauptmann, A. 2014. Instructional videos for unsupervised harvesting and learning of action examples. In *MM*.

Zhang, H.; Yang, Y.; Luan, H.; Yang, S.; and Chua, T.-S. 2014. Start from scratch: Towards automatically identifying, modeling, and naming visual attributes. In *MM*.

Zhao, Q.; Meng, D.; Xu, Z.; Zuo, W.; and Zhang, L. 2014. Robust principal component analysis with complex noise. In *ICML*.

Zhu, C.; Byrd, R. H.; Lu, P.; and Nocedal, J. 1997. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software* 23(4):550–560.