
Self-Paced Learning: an Implicit Regularization Perspective

Yanbo Fan[†] Ran He^{*,†,‡} Jian Liang^{*,†} Bao-Gang Hu[†]

^{*}Center for Research on Intelligent Perception and Computing, CASIA

[†]National Laboratory of Pattern Recognition, CASIA

[‡]Center for Excellence in Brain Science and Intelligence Technology, CAS

{yanbo.fan, rhe, jian.liang, hubg}@nlpr.ia.ac.cn

Abstract

Self-paced learning (SPL) mimics the cognitive mechanism of humans and animals that gradually learns from easy to hard samples. One key issue in SPL is to obtain better weighting strategy that is determined by minimizer function. Existing methods usually pursue this by artificially designing the explicit form of SPL regularizer. In this paper, we focus on the minimizer function, and study a group of new regularizer, named self-paced implicit regularizer that is deduced from robust loss function. Based on the convex conjugacy theory, the minimizer function for self-paced implicit regularizer can be directly learned from the latent loss function, while the analytic form of the regularizer can be even known. A general framework (named SPL-IR) for SPL is developed accordingly. We demonstrate that the learning procedure of SPL-IR is associated with latent robust loss functions, thus can provide some theoretical inspirations for its working mechanism. We further analyze the relation between SPL-IR and half-quadratic optimization. Finally, we implement SPL-IR to both supervised and unsupervised tasks, and experimental results corroborate our ideas and demonstrate the correctness and effectiveness of implicit regularizers.

1 Introduction

Inspired by the learning process and cognitive mechanism of humans and animals, Bengio *et al.* propose a new learning strategy called *curriculum learning* (CL) in [1], which gradually includes more and more hard samples into training process. A curriculum can be seen as a sequence of training criteria. For example, in the training of a shape recognition system, images that exhibit less variability such as squares and circles are considered first, followed by hard shapes like ellipses. The curriculum in CL is usually determined by some certain priors, and thus is problem specific and lacks generalizations. To alleviate this, Kumar *et al.* propose a new learning strategy named self-paced learning (SPL) that incorporates the curriculum updating in the process of model optimization [14]. General SPL model consists of a problem specific weighted loss term on all samples and a SPL regularizer on sample weights. Alternative search strategy (ASS) is generally used for optimization. By gradually increasing the penalty of the SPL regularizer during the optimization, more samples are included into training from easy to hard by a self-paced manner. Due to its ability of avoiding bad local minima and improving the generalization performance, many works have been developed based on SPL [16, 17, 13, 31, 25, 15].

One key issue in SPL is to obtain better weighting strategy that is determined by the minimizer functions, and existing methods usually pursue this by artificially designing the explicit form of SPL regularizers [29, 32, 11, 12]. Some examples are listed in the appendix. Specifically, a definition of self-paced regularizer is given in [11]. Though shown to be effective in many applications experimentally, the underlying working mechanism of SPL is still unclear and is heavily desired for its

future development. One attempt in this aspect is [19], they show that the ASS method used for SPL accords with the *majorization minimization* [26] algorithm implemented on a latent SPL objective, and deduce the latent objective of hard, linear and mixture regularizers.

Considering the crucial role of minimizer function in SPL, we focus on it and study a group of new regularizer (named self-paced implicit regularizer) for SPL based on the convex conjugacy theory. Comparing with existing SPL regularizers, the self-paced implicit regularizer is deduced from robust loss function and its analytic form can be even unknown. Its properties and corresponding minimizer function can be learned from the latent loss function directly. Besides, the proposed self-paced implicit regularizer is independent of the learning objective and thus leads to a general framework (named SPL-IR) for SPL. SPL-IR can be optimized via ASS algorithm. More importantly, we demonstrate that the learning procedure of SPL-IR is indeed associated with latent robust loss functions, thus may provide some theoretical inspirations for its working mechanism (e.g. its robustness to outliers and heavy noise). We further analyze the relations between SPL-IR and half-quadratic (HQ) optimization and provide a group of self-paced implicit regularizer accordingly. Such relations can be beneficial to both SPL and HQ optimization. Finally, we implement SPL-IR to three classical tasks (i.e. matrix factorization, clustering and classification). Experimental results corroborate our ideas and demonstrate the correctness and effectiveness of SPL-IR.

Our work has three main contributions: (1) We propose self-paced implicit regularizer for SPL, and develop a general implicit regularization framework (named SPL-IR) based on it. The self-paced implicit regularizers not only enrich the family of regularizers for SPL but also can provide some inspirations on the working mechanism of SPL. (2) We analyze the connections between SPL-IR and HQ optimization, and provide a group of robust loss function induced self-paced implicit regularizers for SPL-IR accordingly. (3) Experimental results on both supervised and unsupervised tasks corroborate our ideas and demonstrate the correctness and effectiveness of SPL-IR.

2 Preliminaries

2.1 Self-Paced Learning via Explicit Regularizers

Given training dataset $\mathbf{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with n samples, where $\mathbf{x}_i \in R^d$ is the i -th sample, y_i is the optional information according to the learning objective (e.g. y_i can be the label of \mathbf{x}_i in classification model). Let $f(\cdot, \mathbf{w})$ denote the learned model and \mathbf{w} be the model parameter. $L(y_i, f(\mathbf{x}_i, \mathbf{w}))$ is the loss function of i -th sample.

Mimicking the cognitive mechanism of humans and animals, SPL aims to optimize the model from easy to hard samples gradually. The objective of SPL is to jointly optimize model parameter \mathbf{w} and latent sample weights $\mathbf{v} = [v_1, v_2, \dots, v_n]$ via the following minimization problem:

$$\min_{\mathbf{w}, \mathbf{v}} \mathbb{E}(\mathbf{w}, \mathbf{v}; \lambda) = \sum_{i=1}^n v_i L(y_i, f(\mathbf{x}_i, \mathbf{w})) + g(\lambda, v_i), \quad (1)$$

where $g(\lambda, v)$ is called self-paced regularizer and λ is a penalty parameter that controls the learning pace. ASS algorithm is generally used for (1), which alternatively optimizes \mathbf{w} and \mathbf{v} while keeping the other fixed. Specifically, given sample weights \mathbf{v} , the minimization over \mathbf{w} is a weighted loss minimization problem that is independent of regularizer $g(\lambda, v)$; given model parameter \mathbf{w} , the optimal weight of i -th sample is determined by

$$\min_{v_i} v_i L(y_i, f(\mathbf{x}_i, \mathbf{w})) + g(\lambda, v_i). \quad (2)$$

Since $\ell_i = L(y_i, f(\mathbf{x}_i, \mathbf{w}))$ is constant once \mathbf{w} is given, the optimal value of v_i is uniquely determined by the corresponding minimizer function $\sigma(\lambda, \ell_i)$ that satisfies

$$\sigma(\lambda, \ell_i) \ell_i + g(\lambda, \sigma(\lambda, \ell_i)) \leq v_i \ell_i + g(\lambda, v_i), \forall v_i \in [0, 1]. \quad (3)$$

For example, if $g(\lambda, v_i) = -\lambda v_i$ [14], the optimal v_i^* is calculated by

$$v_i^* = \sigma(\lambda, \ell_i) = \begin{cases} 1, & \text{if } \ell_i \leq \lambda \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

By gradually increasing the value of λ , more and more hard samples are included into the training process. Many efforts have been put into the learning of minimizer functions [29, 32, 11, 12, 25], and we name them as SPL with explicit regularizers as they usually require the explicit form of regularizer $g(\lambda, v)$. $\sigma(\lambda, \ell)$ is then derived from the form of $g(\lambda, v)$.

Table 1: Loss function $\phi(\lambda, t)$ and the corresponding minimizer function $\sigma(\lambda, t)$, λ is a hyper-parameter.

	Huber	Cauchy	L1-L2	Welsch
Loss function $\phi(\lambda, t)$	$\begin{cases} t^2/2, & t \leq \lambda \\ \lambda t - \frac{\lambda^2}{2}, & t > \lambda \end{cases}$	$\lambda^2 \log(1 + (t/\lambda)^2)$	$\sqrt{\lambda + t^2} - 1$	$\lambda^2(1 - \exp(-\frac{t^2}{\lambda^2}))$
Minimizer function $\sigma(\lambda, t)$	$\begin{cases} 1, & t \leq \lambda \\ \lambda/ t , & t > \lambda \end{cases}$	$2/(1 + (t/\lambda)^2)$	$1/\sqrt{\lambda + t^2}$	$2 \exp(-\frac{t^2}{\lambda^2})$

2.2 Half-Quadratic Optimization

Half-quadratic optimization [21, 5, 4] is a commonly used optimization method that based on the convex conjugacy theory. It tries to solve a nonlinear objective function via optimizing a series of half-quadratic reformulation problems iteratively [7, 9, 8, 6, 30].

Given a differentiable function $\phi(t) : R \rightarrow R$, if $\phi(t)$ further satisfies the conditions of the multiplicative form of HQ optimization in [20], the following equation holds for any fixed t ,

$$\phi(t) = \inf_{p \in R_+} \left\{ \frac{1}{2}pt^2 + \psi(p) \right\}, \quad (5)$$

where $\psi(p)$ is the dual potential function of $\phi(t)$ and $R_+ = \{t|t \geq 0\}$. $\psi(p)$ is convex and reads

$$\psi(p) = \sup_{t \in R_+} \left\{ -\frac{1}{2}pt^2 + \phi(t) \right\}, \quad (6)$$

More analysis about $\phi(t)$ and $\psi(p)$ refers to [21]. The optimal p^* that minimize (5) is uniquely determined by the corresponding minimizer function $\delta(t)$, which is derived from convex conjugacy and is only relative to function $\phi(t)$. For each t , $\delta(t)$ is such that

$$\frac{1}{2}\delta(t)t^2 + \psi(\delta(t)) \leq \frac{1}{2}pt^2 + \psi(p), \quad \forall p \in R_+. \quad (7)$$

The optimization of $\phi(t)$ can be done via iteratively minimizing t and p in (5). One only needs to focus on $\phi(t)$ and its corresponding minimizer function $\delta(t)$ in HQ optimization, and the analytical form of the dual potential function $\psi(p)$ can be even unknown.

3 The Proposed Method

In this section, we first give the definition of the proposed self-paced implicit regularizer and derive its minimizer function based on convex conjugacy. Then we develop a general self-paced learning framework, named SPL-IR, based on implicit regularization. Finally, we analyze the relations between SPL-IR and HQ optimization.

3.1 Self-Paced Implicit Regularizer

Based on our above analysis of SPL, we define the self-paced implicit regularizer as follows,

Definition 1. Self-Paced Implicit Regularizer. A self-paced implicit regularizer $\psi(\lambda, v)$ is defined as the dual potential function of a robust loss function $\phi(\lambda, t)$, and satisfies

1. $\phi(\lambda, t) = \min_{v \geq 0} vt + \psi(\lambda, v)$;
2. $\sigma(\lambda, t)$ is the minimizer function of $\phi(\lambda, t)$ that satisfies $\sigma(\lambda, t)t + \psi(\lambda, \sigma(\lambda, t)) \leq vt + \psi(\lambda, v)$, $\forall v \in R_+$;
3. $\sigma(\lambda, t)$ is non-negative and up-bounded, $\forall t \in R_+$;
4. $\sigma(\lambda, t)$ is monotonically decreasing w.r.t. t , $\forall t \in R_+$;
5. $\sigma(\lambda, t)$ is monotonous w.r.t. $\lambda \in R_+$;

where λ is a hyper-parameter and it is the same in $\phi(\lambda, t)$, $\psi(\lambda, v)$ and $\sigma(\lambda, t)$. λ is considered to be fixed in the first four conditions.

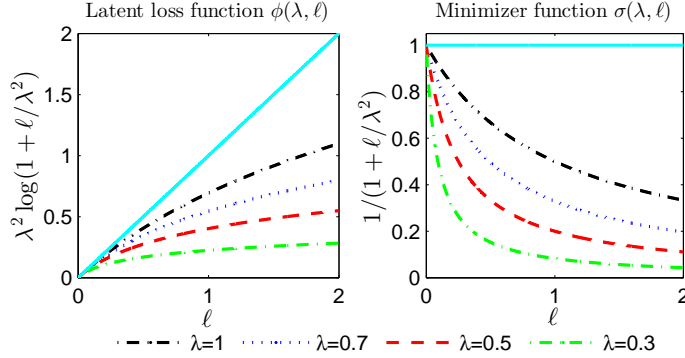


Figure 1: Example of latent loss function and its corresponding minimizer function in Definition 1. The x-axis refers to original loss ℓ . The solid lines are given for comparison, it is $y = x$ in left figure, and $y = 1$ in right one.

Proposition 1 For any fixed λ , if $\phi(\lambda, t)$ in Definition 1 further satisfies the conditions referred in [20], its minimizer function $\sigma(\lambda, t)$ is uniquely determined by $\phi(\lambda, t)$ and the analytic form of the dual potential function $\psi(\lambda, v)$ can be even unknown during the optimization.

The proof of Proposition 1 is given in the appendix. According to Definition 1, the self-paced implicit regularizer is derived from robust loss function. Its properties can be learned from both $\psi(\lambda, v)$ and the latent loss function $\phi(\lambda, t)$. The corresponding minimizer function $\sigma(\lambda, t)$ can be learned from $\phi(\lambda, t)$ directly. During the optimization, the optimal v^* is determined by $\sigma(\lambda, t)$ and the analytic form of $\psi(\lambda, v)$ can be even unknown, hence $\psi(\lambda, v)$ is named self-paced implicit regularizer. Besides, the last three conditions in Definition 1 are required for SPL regimes. Specifically, let t denote the sample loss, condition 4 indicates that the model is likely to select easy samples (with smaller losses) in favor of hard samples (with larger losses) for a fixed λ , and condition 5 makes sure that we can incorporate more and more samples through turning parameter λ .

Besides, Jiang *et al.* have given a definition of self-paced regularizer and derived necessary conditions of the regularizer and the corresponding minimizer function for SPL in [11]. However, it is still nontrivial to design self-paced regularizers or analyze their properties accordingly. The self-paced implicit regularizer $\psi(\lambda, v)$ defined here is derived from robust loss function $\phi(\lambda, t)$. By establishing the relations between $\phi(\lambda, t)$ and $\psi(\lambda, v)$, we can analyze their working mechanisms as well as develop new SPL regularizers based on the development of robust loss functions. Moreover, the properties of $\psi(\lambda, v)$ and its corresponding minimizer function $\sigma(\lambda, t)$ can be learned from $\phi(\lambda, t)$.

3.2 Self-Paced Learning via Implicit Regularizers

We can develop an implicit regularization framework for SPL based on the proposed self-paced implicit regularizer. By substituting the regularization term $g(\lambda, v)$ in (1) with a self-paced implicit regularizer $\psi(\lambda, v)$ given in Definition 1, we obtain the following SPL-IR problem,

$$\min_{\mathbf{w}, \mathbf{v}} \mathbb{E}(\mathbf{w}, \mathbf{v}; \lambda) = \sum_{i=1}^n v_i L(y_i, f(\mathbf{x}_i, \mathbf{w})) + \psi(\lambda, v_i). \quad (8)$$

It can be solved via ASS algorithm, which alternatively optimizes \mathbf{w} and \mathbf{v} while keeping the other fixed. However, different from existing SPL regularizers, the analytic form of $\psi(\lambda, v)$ in (8) can be unknown and the optimal \mathbf{v}^* is determined by the corresponding minimizer function given in Definition 1. The optimization procedure of (8) is described in Algorithm 1. Model (8) is called an implicit regularization framework since it does not require the explicit form of $\psi(\lambda, v)$. The benefit of implicit regularization has been analyzed in [18, 22].

An insightful phenomenon is that the learning procedure of SPL-IR is actually associated with certain latent loss functions. For example, for a certain implicit regularizer and its corresponding minimizer function $v_i^* = \sigma(\lambda, \ell_i) = 1/(1 + \ell_i/\lambda^2)$ in Algorithm 1 (where $\ell_i = L(y_i, f(\mathbf{x}_i, \mathbf{w}^*))$), one is actually minimizing a latent robust function $\sum_{i=1}^n \lambda^2 \log(1 + \ell_i/\lambda^2)$ during each round. Figure 1 gives a graphical illustration. The latent loss function $\phi(\lambda, \ell)$ can be considered to carry out a meaningful transformation on original loss ℓ . When ℓ is larger than a certain threshold, $\phi(\lambda, \ell)$

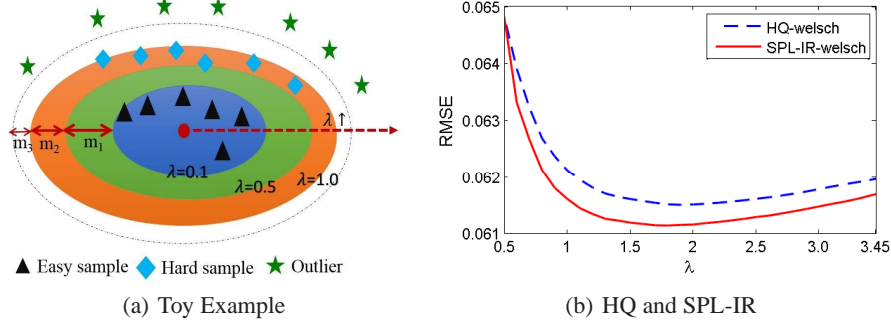


Figure 2: In (a), training samples are roughly divided into three types: easy samples \blacktriangle , hard samples \blacklozenge and outliers \star . λ is usually fixed in HQ methods (e.g. $\lambda = 0.5$), hence some samples may be discarded incorrectly. In contrast, SPL-IR can gradually incorporate more samples from easy to hard (i.e. λ grows iteratively). (b) demonstrates the performances of HQ and SPL-IR methods on a synthetic matrix factorization dataset, Welsch minimizer function is adopted for both methods. For HQ-welsch, standard HQ algorithm [21] is implemented with each λ independently. More details refer to Section 3.3 and 4.1.

becomes a constant and its corresponding minimizer function $\sigma(\lambda, \ell)$ becomes zero, hence the related sample is not considered for optimization. Through this, it can suppress the influence of hard samples (refer to larger ℓ) while retaining that of easy samples (refer to smaller ℓ). This may also provide some inspirations on the robustness of SPL-IR to outliers and heavy noise as they can usually cause larger losses. More specifically, starting with a small λ (e.g. 0.3), only a small part of samples with very small losses will be involved (they are considered to contain reliable information). As λ increases, the suppressing effect of $\phi(\lambda, \ell)$ on larger losses becomes weaker and their corresponding weights increase, consequently more and more hard samples with larger losses (may also contain more knowledge) are involved into training process. While gradually incorporating these knowledge, the model becomes stronger and stronger. The learning procedure of some existing regularizers like hard and linear [19] can also be explained under the framework of SPL-IR.

SPL-IR in (8) is considered as a general SPL framework from two aspects: firstly, $\psi(\lambda, v)$ represents a spectrum of self-paced implicit regularizer that is developed based on robust loss function and convex conjugacy theory; secondly, $\psi(\lambda, v)$ is independent of specific model objective $L(y_i, f(\mathbf{x}_i, \mathbf{w}))$ and thus can be used in various applications. Besides, standard ASS strategy is used for both SPL with explicit regularizer (model (1)) and SPL-IR (model (8)). It includes a weighted loss minimization step and a weight updating step at each iteration, and the time overhead is mainly in the former step. Hence for a specific loss function $L(y_i, f(\mathbf{x}_i, \mathbf{w}))$ and a fixed number of iteration, the time complexities of SPL with explicit regularizer and SPL-IR is in the same order of magnitude.

3.3 SPL-IR and Half-Quadratic Optimization

We can develop new self-paced implicit regularizers based on the development of robust loss functions. Specifically, we analyze the relations between SPL-IR and HQ optimization and provide several self-paced implicit regularizers accordingly. For better demonstration, we first give an equivalent quadratic form definition of self-paced implicit regularizer,

Definition 2 (Quadratic Form). *Self-Paced Implicit Regularizer.* A self-paced implicit regularizer $\psi(\lambda, v)$ is defined as the dual potential function of a robust loss function $\phi(\lambda, t)$, and satisfies

1. $\phi(\lambda, t) = \min_{v \geq 0} \frac{1}{2} vt^2 + \psi(\lambda, v)$;
2. $\sigma(\lambda, t)$ is the minimizer function of $\phi(\lambda, t)$ and satisfies $\frac{1}{2}\sigma(\lambda, t)t^2 + \psi(\lambda, \sigma(\lambda, t)) \leq \frac{1}{2}vt^2 + \psi(\lambda, v)$, $\forall v \in R_+$;
3. $\sigma(\lambda, t)$ is non-negative and up-bounded, $\forall t \in R_+$;
4. $\sigma(\lambda, t)$ is monotonically decreasing w.r.t. t , $\forall t \in R_+$;
5. $\sigma(\lambda, t)$ is monotonous w.r.t. $\lambda \in R_+$;

where λ is a hyper-parameter and it is the same in $\phi(\lambda, t)$, $\psi(\lambda, v)$ and $\sigma(\lambda, t)$. λ is considered to be fixed in the first four conditions.

Algorithm 1: Self-Paced Learning via Implicit Regularizers

Input: Input dataset $\mathbf{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$, step size $\mu > 1$.**Output:** Model parameter \mathbf{w} .

- 1: Initialize sample weights \mathbf{v}^* and parameter λ ;
 - 2: **repeat**
 - 3: Update $(\mathbf{w}^*, \mathbf{v}^*) = \arg \min_{\mathbf{w}, \mathbf{v}} \mathbb{E}(\mathbf{w}, \mathbf{v}; \lambda)$ by using ASS algorithms, \mathbf{v} is iteratively optimized by the corresponding minimizer function σ ;
 - 4: Monotone increase (or decrease) λ by step-size μ ;
 - 5: **until** convergence.
 - 6: **return** \mathbf{w}^*
-

Table 2: Numerical results of L_1 -norm MF problem with L_2 -norm regularization. The best results are highlighted in bold.

Method	PRMF	SPL-hard	SPL-mixture	SPL-IR-huber	SPL-IR-L1-L2	SPL-IR-cauchy	SPL-IR-welsch
RMSE	0.1528	0.0949	0.0625	0.0627	0.0650	0.0620	0.0596
MAE	0.0994	0.0672	0.0475	0.0476	0.0493	0.0472	0.0455

The equivalency of Definition 1 and Definition 2 is shown in the appendix. Seen from Definition 2, there is a close relationship between self-paced implicit regularizer and the dual potential function defined in HQ reformulation (5). Apparently, the dual potential function in (5) and the minimizer function in (7) satisfy the first two conditions in Definition 2, and self-paced implicit regularizer imposes further constraints on the minimizer function $\sigma(\lambda, t)$ for the regimes of SPL. Many loss functions and their corresponding minimizer functions in multiplicative form of HQ have been developed (some of them are tabulated in Table 1). It is easy to verify that the functions in Table 1 satisfy all the conditions in Definition 2, hence they can be adjusted for self-paced implicit regularizers. The loss functions in Table 1 are well defined and have proven to be effective in many areas [9]. Meanwhile, though self-paced implicit regularizer can be developed from HQ optimization, their optimization procedures are quite different. In HQ, one mainly focuses on the minimization of loss function $\phi(\lambda, t)$ and hyper-parameter λ is predetermined and fixed during the optimization. While aiming to gradually optimize from easy to hard samples, SPL-IR uses the right-hand side $vt^2/2 + \psi(\lambda, v)$ to model problems and one key concern is the weighting strategy that determined by the minimizer function $\sigma(\lambda, t)$. Besides, in order to gradually increase samples, λ is updated stage by stage in SPL-IR.

Figure 2 gives an intuitive interpretation. If we set $t_i = \sqrt{L(y_i, f(\mathbf{x}_i, \mathbf{w}^*))}$ and use the minimizer function of Welsch given in Table 1 for weight updating in Algorithm 1, model (8) can be considered to sequentially optimize a group of Welsch loss functions with monotonically increasing λ . Hence SPL-IR is able to gradually optimize from easy to hard samples while incorporating the good properties of robust Welsch functions. On the other hand, for HQ optimization, λ is predefined and fixed during the whole optimization. Hence its performance may be largely influenced by the selection of λ . For example, when λ is somehow small (e.g. $\lambda < 1$ in Figure 2(b)), some hard samples will be simply considered as outliers and discarded. From the comparisons in Figure 2(b), we can find that SPL-IR can always outperform HQ for every λ .

4 Experiments

To illustrate the correctness and effectiveness of the developed SPL-IR model, we apply it to three classical tasks: matrix factorization, clustering and classification. Experimental results demonstrate that the proposed self-paced implicit regularizers outperform baseline algorithms and achieve comparable or even better performance comparing to the artificially designed SPL regularizers.

There are two hyper-parameter (λ, μ) that need to be tuned in Algorithm 1. We follow a standard setting in SPL [14] for all our experiments. That is, λ is initialized to obtain about half samples, then it is iteratively updated to involve more and more samples gradually. The practical updating direction depends on the specific minimizer function. For functions given in Table 1, $\lambda_{T+1} = \lambda_T/\mu$ for L1-L2 while $\lambda_{T+1} = \lambda_T * \mu$ for Huber, Cauchy and Welsch, where $\mu > 1$ is a step factor and T

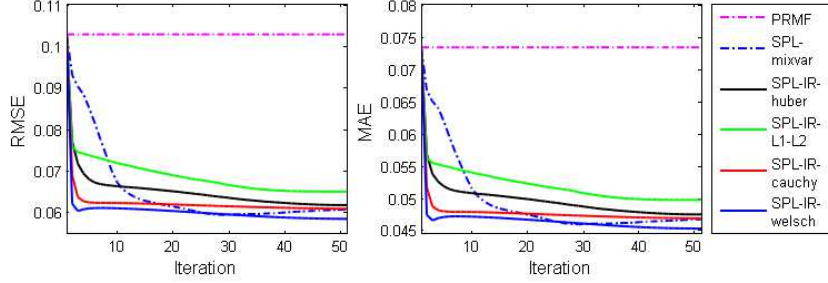


Figure 3: Tendency curves of RMSE and MAE w.r.t. the iterations.

Table 3: Clustering performance on the Handwritten Digit dataset. The best results are highlighted in bold.

Method	ACC	NMI	AR	F-score	Purity
FOU	0.612(0.066)	0.628(0.029)	0.484(0.049)	0.539(0.043)	0.645(0.051)
FAC	0.588(0.044)	0.597(0.017)	0.453(0.031)	0.512(0.027)	0.631(0.032)
KAR	0.734(0.062)	0.730(0.030)	0.634(0.055)	0.672(0.049)	0.767(0.048)
MOR	0.415(0.014)	0.500(0.003)	0.295(0.004)	0.374(0.003)	0.475(0.004)
PIX	0.677(0.059)	0.701(0.031)	0.585(0.050)	0.629(0.045)	0.711(0.047)
ZER	0.524(0.033)	0.504(0.016)	0.369(0.024)	0.434(0.021)	0.551(0.022)
Con-MC	0.775(0.078)	0.773(0.037)	0.690(0.066)	0.722(0.058)	0.802(0.059)
SPL-hard	0.821(0.059)	0.758(0.029)	0.709(0.050)	0.739(0.044)	0.834(0.045)
SPL-mixture	0.845(0.068)	0.812(0.030)	0.763(0.057)	0.787(0.051)	0.861(0.050)
MSPL	0.840(0.070)	0.806(0.035)	0.751(0.064)	0.776(0.057)	0.854(0.054)
SPL-IR-huber	0.843(0.070)	0.810(0.035)	0.756(0.064)	0.781(0.057)	0.858(0.053)
SPL-IR-L1-L2	0.835(0.068)	0.801(0.034)	0.743(0.061)	0.769(0.054)	0.849(0.052)
SPL-IR-cauchy	0.845(0.071)	0.814(0.035)	0.762(0.064)	0.786(0.057)	0.861(0.053)
SPL-IR-welsch	0.862(0.071)	0.833(0.035)	0.790(0.064)	0.812(0.057)	0.878(0.053)

is an iteration number. μ is empirically set to 1.05 in our experiments. Similar settings are adjusted for the competing SPL regularizers, including SPL-hard [14] and SPL-mixture [32].

4.1 Matrix Factorization

Matrix factorization (MF) is one of the fundamental problems in machine learning and data mining. It aims to factorize an $m \times n$ data matrix \mathbf{Y} into two smaller factors $\mathbf{U} \in R^{m \times r}$ and $\mathbf{V} \in R^{n \times r}$, where $r \ll \min(m, n)$, such that \mathbf{UV}^T is possibly close to \mathbf{Y} . MF has been successfully implemented in many applications, such as collaborative filtering [24].

Here we consider the MF problem on synthetic dataset. Specifically, the data used here is generated as follows: two matrices \mathbf{U} and \mathbf{V} , both of which are of size 100×4 , are first randomly generated with each entry drawn from the Gaussian distribution $\mathcal{N}(0, 1)$, leading to a ground truth rank-4 matrix $\mathbf{Y}_0 = \mathbf{UV}^T$. Then we randomly choose 40% of the entries and treat them as missing data. Another 20% of the entries are randomly selected and added to uniform noise on $[-20, 20]$, and the rest are perturbed with Gaussian noise drawn from $\mathcal{N}(0, 0.1^2)$. Similar to [32], we consider L_1 -norm MF problem with L_2 -norm regularization, and the baseline algorithm is PRMF [27]. We modify it with different SPL regularizers for comparison. Two commonly used metrics are adopted here: (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 $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ denote the outputs of MF algorithms. All the algorithms are implemented with 50 realizations and their mean values are reported.

Table 2 tabulates their numerical results. All SPL-IR algorithms obtain performance improvements over baseline algorithm PRMF, which shows the benefits of SPL regimes. Comparing among different SPL regularizers, the results of proposed self-paced implicit regularizers are comparable to or even better than that of mixture and hard schemes, especially for SPL-IR with welsch regularizer. These demonstrate the correctness and effectiveness of the proposed self-paced implicit regularizer. Figure 3 further plots the tendency curves of RMSE and MAE with different self-paced implicit regularizers and mixture regularizer for better understanding, the results of PRMF are also reported as a baseline. The performances of all implicit regularizers improve rapidly for the first few iterations

Table 4: Statistical Information of Databases.

Dataset	#.Category	#.Instance	#.Feature
Breast	2	569	30
Spambase	2	4601	57
Svmguide1	2	7089	4

Table 5: Classification accuracy (%).

Without Label Noise							
Method	LR	SPL-hard	SPL-mixture	SPL-IR-huber	SPL-IR-L1-L2	SPL-IR-cauchy	SPL-IR-welsch
Breast	97.36(2.22)	97.54(2.22)	98.25(1.65)	98.77(1.19)	97.90(1.79)	98.42(1.54)	98.25(1.65)
Spambase	92.35(1.47)	92.63(1.08)	92.83(1.44)	93.05(1.25)	93.00(1.36)	93.09(1.41)	93.13(1.34)
Svmguide1	95.39(0.95)	95.39(0.95)	95.51(1.04)	95.57(0.95)	95.57(1.10)	95.65(1.01)	95.68(0.90)
With 20% Random Label Noise							
Method	LR	SPL-hard	SPL-mixture	SPL-IR-huber	SPL-IR-L1-L2	SPL-IR-cauchy	SPL-IR-welsch
Breast	92.08(2.96)	96.13(2.15)	96.66(2.12)	96.84(2.33)	94.72(2.89)	97.54(1.90)	97.89(1.63)
Spambase	89.28(1.66)	89.81(1.61)	90.76(1.82)	90.92(1.65)	90.09(1.65)	90.85(1.55)	91.37(1.37)
Svmguide1	91.52(0.65)	92.72(1.12)	93.81(0.79)	93.54(0.75)	92.83(0.71)	93.88(1.05)	94.37(0.90)

as more and more easy samples are likely to be involved in these phases. With the increasing of the iterations, the improvements become steady as some hard instances or outliers are included.

4.2 Multi-view Clustering

Multi-view clustering aims to group data with multiple views into their underlying classes [28]. Most existing multi-view clustering algorithms fit a non-convex model and may be stuck in bad local minima. To alleviate this, Xu *et al.* propose a multi-view self-paced learning algorithm (MSPL) that considers the learnability of both samples and views and achieves promising results in [29]. Here we simply modified their MSPL model with different SPL regularizers for comparison. The UCI Handwritten Digit dataset¹ is used in this experiment. It consists of 2,000 handwritten digits classified into ten categories (0-9). Each instance is represented in terms of the following six kinds of features (or views): Fourier coefficients of the character shapes (FOU), profile correlations (FAC), Karhunen-Love coefficients (KAR), pixel averages in 2 x 3 windows (PIX), Zernike moments (ZER), and morphological features (MOR). Here we make use of all the six views for all the comparing algorithms. The baseline algorithms are standard k-means on each single view’s representation and Con-MC (the features are concatenated on all views firstly, and then standard k-means is applied).

Five commonly used metrics are adopted to measure the clustering performances: clustering accuracy (ACC), normalized mutual information (NMI), F-score, Purity, and adjusted rand index (AR) [10]. Higher value indicates better performance for all the metrics. All algorithms are implemented 20 times and both mean values and standard derivations are reported. Table 3 tabulates their numerical results. It can be seen that all the multi-view algorithms obtain significant improvements over single-view ones, which demonstrates the benefits of integrating information from different views. More importantly, comparing to Con-MC, the SPL-IR algorithms can further improve the performance by gradually optimizing from easy to hard samples and avoiding bad local minima. The proposed self-paced implicit regularizers are comparable to or even better than the compared SPL regularizers.

4.3 Classification

The proposed self-paced implicit regularizers can be flexible implemented to supervised tasks. Here we conduct a binary classification task. Specifically, we utilize the L2-regularized Logistic Regression (LR) model as our baseline, and incorporate it with different SPL regularizers for comparison. Liblinear [3] is used as the solver of LR. Three real-world databases are considered: Breast¹, Spambase¹ and Svmguide1 [2]. Their statistical information is summarized in Table 4. For each dataset,

¹<https://archive.ics.uci.edu/ml/datasets>

we consider it without additional noise and with 20% random label noise, respectively. The 20% random label noise means we randomly select 20% samples from training data and reversal their labels (change positive to negative, and vice-versa). We use 10-fold cross validation for all the databases, and report both their mean values and their standard derivations.

Classification accuracy is used for performance measure. Table 5 reports their numerical results. For both situations, SPL-IR algorithms can get performance improvements over original LR method to some extent. Moreover, when adding random label noise, the performance of original LR degenerates a lot, while the SPL algorithms can still obtain relatively high performance, especially for SPL-IR with welsch regularizer. This corroborates our analysis about the robustness of SPL-IR to outliers and heavy noise.

5 Conclusions

In this paper, we study a group of new regularizer, named self-paced implicit regularizer for SPL based on the convex conjugate theory. The self-paced implicit regularizer is derived from robust loss function and its analytic form can be even unknown. Its properties and the corresponding minimizer function can be learned from the latent loss function directly. We then develop a general SPL framework (SPL-IR) based on it. We further demonstrate that the learning procedure of SPL-IR is actually associated with certain latent robust loss functions, thus may provide some theoretical inspirations on the working mechanisms of SPL-IR (such as the robustness to outliers or heavy noise). We later analyze the relations between SPL-IR and HQ optimization and develop a group of self-paced implicit regularizer accordingly. Experimental results on both supervised and unsupervised tasks demonstrate the correctness and effectiveness the proposed self-paced implicit regularizer.

6 Appendix

6.1 Proof of Proposition 1

Proof. The proof sketch is similar to that in [20]. For ease of representation, we omit λ and use $\phi(t)$, $\psi(v)$ and $\sigma(t)$ for short. Some fundamental assumptions about $\phi(t)$ are: **H1**: $\phi : R_+ \rightarrow R$ is increasing with $\phi \not\equiv 0$ and $\phi(0) = 0$; **H2**: $\phi(t)$ is C^1 and concave; **H3**: $\lim_{t \rightarrow \infty} \phi(t)/t = 0$.

Put $\theta(t) = -\phi(t)$, then θ is convex by H2. Its convex conjugate is $\theta^*(v) = \sup_{t \geq 0} \{vt - \theta(t)\}$. By the Fenchel-Moreau theorem [23], the convex conjugate of θ^* is θ , that is $\theta(t) = (\theta^*)^*(t) = \sup_{v \leq 0} \{vt - \theta^*(v)\} = -\inf_{v \geq 0} \{vt + \theta^*(-v)\}$. Thus we have

$$\psi(v) = \theta^*(-v) = \sup_{t \geq 0} \{-vt - \theta(t)\} = \sup_{t \geq 0} \{-vt + \phi(t)\}. \quad (9)$$

$$\phi(t) = -\theta(t) = \inf_{v \geq 0} \{vt + \theta^*(-v)\} = \inf_{v \geq 0} \{vt + \psi(v)\}. \quad (10)$$

Then the problem becomes how to achieve the supremum in (9) jointly with the infimum in (10). For any $\hat{v} > 0$, define $f_{\hat{v}} : R_+ \rightarrow R$ by $f_{\hat{v}}(t) = \hat{v}t + \theta(t)$, then we have $\psi(\hat{v}) = -\inf_{t \geq 0} f_{\hat{v}}(t)$ from (9). According to H1-H3, $f_{\hat{v}}$ is convex with $f_{\hat{v}}(0) = 0$ and $\lim_{t \rightarrow +\infty} f_{\hat{v}}(t) = +\infty$. Thus $f_{\hat{v}}$ can reach its unique minimum at a $\hat{t} \geq 0$, and $\psi(\hat{v}) = -\hat{v}\hat{t} + \phi(\hat{t})$ from (9). Hence equivalently the infimum in (10) is reached at \hat{v} as $\phi(\hat{t}) = \hat{v}\hat{t} + \psi(\hat{v})$. Then we have $\hat{v} = \sigma(t) = -\theta'(t) = \phi'(t)$. Thus the optimal v is uniquely determined by the minimizer function $\sigma(t)$ that is derived from $\phi(t)$. The analytic form of the dual potential function $\psi(v)$ could be unknown during the optimization. The proof is then completed.

6.2 Definition 1 and Definition 2

To show the equivalency of Definition 1 and Definition 2 in the main body, we first give the following proposition about Definition 2.

Proposition 2 *For any fixed λ , if $\phi(\lambda, t)$ in Definition 2 further satisfies the conditions referred in [20], its minimizer function $\sigma(\lambda, t)$ is uniquely determined by $\phi(\lambda, t)$ and the analytic form of $\psi(\lambda, v)$ can be even unknown during the optimization.*

Proof. The proof sketch is similar to that in [20]. For ease of representation, we omit λ and use $\phi(t)$, $\psi(v)$ and $\sigma(t)$ for short. Some fundamental assumptions about $\phi(t)$ are: **H1**: $\phi : R_+ \rightarrow R$ is increasing with $\phi \not\equiv 0$ and $\phi(0) = 0$; **H2**: $t \rightarrow \phi(\sqrt{t})$ is concave; **H3**: $\phi(t)$ is C^1 ; **H4**: $\lim_{t \rightarrow \infty} \phi(t)/t^2 = 0$.

Put $\theta(t) = -\phi(\sqrt{t})$, then θ is convex by H2. Its convex conjugate is $\theta^*(v) = \sup_{t \geq 0} \{vt - \theta(t)\}$. By the Fenchel-Moreau theorem [23], the convex conjugate of θ^* is θ , that is $\theta(t) = (\theta^*)^*(t) = \sup_{v \leq 0} \{vt - \theta^*(v)\} = -\inf_{v \geq 0} \{vt + \theta^*(-v)\}$. Define $\psi(v) = \theta^*(-\frac{1}{2}v)$, we have

$$\psi(v) = \sup_{t \geq 0} \left\{ -\frac{1}{2}vt - \theta(t) \right\} = \sup_{t \geq 0} \left\{ -\frac{1}{2}vt^2 + \phi(t) \right\}. \quad (11)$$

$$\phi(t) = -\theta(t^2) = \inf_{v \geq 0} \{vt^2 + \theta^*(-v)\} = \inf_{v \geq 0} \left\{ \frac{1}{2}vt^2 + \psi(v) \right\}. \quad (12)$$

Then the problem becomes how to achieve the supremum in (11) jointly with the infimum in (12). For any $\hat{v} > 0$, define $f_{\hat{v}} : R_+ \rightarrow R$ by $f_{\hat{v}}(t) = \frac{1}{2}\hat{v}t + \theta(t)$, then we have $\psi(\hat{v}) = -\inf_{t \geq 0} f_{\hat{v}}(t)$ from (11). According to H1-H4, $f_{\hat{v}}$ is convex with $f_{\hat{v}}(0) = 0$ and $\lim_{t \rightarrow +\infty} f_{\hat{v}}(t) = +\infty$. Thus $f_{\hat{v}}$ can reach its unique minimum at a $\hat{t} \geq 0$, and $\psi(\hat{v}) = -\frac{1}{2}\hat{v}\hat{t}^2 + \phi(\hat{t})$ from (11). Hence equivalently the infimum in (12) is reached at \hat{v} as $\phi(\hat{t}) = \frac{1}{2}\hat{v}\hat{t}^2 + \psi(\hat{v})$. Then we have $\hat{v} = \sigma(\hat{t}) = -2\theta'(\hat{t}^2) = \phi'(\hat{t})/\hat{t}$. Thus the optimal v is uniquely determined by the minimizer function $\sigma(t)$ that is only related to $\phi(t)$. The analytic form of the dual potential function $\psi(v)$ could be unknown during the optimization. The proof is then completed.

Denote $\ell_i = L(y_i, f(\mathbf{x}_i, \mathbf{w}))$ and rewrite model (8) in the main body as

$$\min_{\mathbf{w}, \mathbf{v}} \mathbb{E}(\mathbf{w}, \mathbf{v}; \lambda) = \sum_{i=1}^n v_i (\sqrt{\ell_i})^2 + \psi(\lambda, v_i). \quad (13)$$

If we adopt $\psi(\lambda, v_i)$ with an implicit regularizer given in Definition 2 and use $v_i^* = \frac{1}{2}\sigma(\lambda, \sqrt{\ell_i})$, where $\sigma(\lambda, \sqrt{\ell_i})$ is the minimizer function in Definition 2, model (13) is optimizing a latent loss function $\sum_{i=1}^n \phi(\lambda, \sqrt{\ell_i})$ equivalently.

Now we demonstrate the equivalency of Definition 1 and Definition 2 in the main body. For easy of representation, we omit λ , and use $\{\phi_1(t), \psi_1(v), \sigma_1(t)\}$ and $\{\phi_2(t), \psi_2(v), \sigma_2(t)\}$ to refer to the functions in Definition 1 and Definition 2, respectively. Considering a simplified model

$$\min_{\mathbf{w}, v} vL(y, f(\mathbf{x}, \mathbf{w})) + \psi(v). \quad (14)$$

Denote $\ell = L(y, f(\mathbf{x}, \mathbf{w}))$. We show that for a same implicit regularizer $\psi(v) = \psi_1(v) = \psi_2(v)$, the optimal v^* and the latent loss function of model (14) derived from Definition 1 and Definition 2 are the same. Specifically, let $\psi_1(v) = \psi_2(v) = \sup_{t \geq 0} \{-vt + \phi_1(t)\}$ (where $\phi_1(t)$ satisfies conditions H1-H3 of Proposition 1 in the main body), it is easy to verify that its corresponding latent loss function is $\phi_1(\ell)$ and optimal $v^* = \sigma_1(\ell) = \phi_1'(\ell)$ according to Definition 1 and Proposition 1. Meanwhile, we have $\psi_2(v) = \sup_{t \geq 0} \{-vt + \phi_1(t)\} = \sup_{t \geq 0} \{-vt^2 + \phi_2(t)\}$, where $\phi_2(t) = \phi_1(t^2)$. Then model (14) can be considered to optimize a latent loss function $\phi_2(\sqrt{\ell}) = \phi_1(\ell)$ and the optimal $v^* = \frac{1}{2}\sigma_2(\sqrt{\ell}) = \phi_1'(\ell)$ according to Definition 2 and Proposition 2. Thus we show the equivalency of Definition 1 and Definition 2.

6.3 Self-Paced Regularizer

Similar definitions of self-paced regularizer (or self-paced function) have been proposed in [13, 32, 11]. The definition in [32] is shown below.

Definition 3 (Self-Paced Regularizer) [32]: Suppose that v is a weight variable, ℓ is the loss, and λ is the learning pace parameter. $g(\lambda, v)$ is called self-paced regularizer, if

1. $g(\lambda, v)$ is convex with respect to $v \in [0, 1]$;
2. $v^*(\lambda, \ell)$ is monotonically decreasing w.r.t. ℓ , and it holds that $\lim_{\ell \rightarrow 0} v^*(\lambda, \ell) = 1$, $\lim_{\ell \rightarrow \infty} v^*(\lambda, \ell) = 0$;

3. $v^*(\lambda, \ell)$ is monotonically increasing w.r.t. λ , and it holds that $\lim_{\lambda \rightarrow 0} v^*(\lambda, \ell) = 0$, $\lim_{\lambda \rightarrow \infty} v^*(\lambda, \ell) \leq 1$;

where $v^*(\lambda, \ell) = \arg \min_{v \in [0, 1]} v\ell + g(\lambda, v)$.

Table 6: Recently proposed self-paced regularizers $g(\lambda, v)$ and their corresponding $v^*(\lambda, \ell)$

	$g(\lambda, v)$	$v^*(\lambda, \ell)$
Kumar <i>et al.</i> [14]	$-\lambda \sum_{i=1}^n v_i, \lambda > 0$	$\begin{cases} 1, & \ell_i < \lambda \\ 0, & \text{otherwise} \end{cases}$
Jiang <i>et al.</i> [11, 13]	$\frac{1}{2}\lambda \sum_{i=1}^n (v_i^2 - 2v_i), \lambda > 0$	$\begin{cases} 1 - \frac{1}{\lambda}\ell_i, & \ell_i < \lambda \\ 0, & \text{otherwise} \end{cases}$
Jiang <i>et al.</i> [11, 13]	$\sum_{i=1}^n (\zeta v_i - \frac{\zeta^{v_i}}{\log \zeta}),$ $\zeta = 1 - \lambda, 0 < \lambda < 1$	$\begin{cases} \frac{1}{\log \zeta} \log(\ell_i + \zeta), & \ell_i < \lambda \\ 0, & \text{otherwise} \end{cases}$
Jiang <i>et al.</i> [11, 13]	$-\zeta \sum_{i=1}^n \log(v_i + \frac{1}{\lambda_1}\zeta),$ $\zeta = \frac{\lambda_1 \lambda_2}{\lambda_1 - \lambda_2}, \lambda_1 > \lambda_2 > 0$	$\begin{cases} 1, & \ell_i \leq \lambda_2 \\ \frac{(\lambda_1 - \ell_i)\zeta}{\ell_i \lambda_1}, & \lambda_2 < \ell_i < \lambda_1 \\ 0, & \ell_i \geq \lambda_1 \end{cases}$
Jiang <i>et al.</i> [12]	$-\lambda \sum_{i=1}^n v_i - \gamma \ \mathbf{v}\ _{2,1}, \lambda > 0, \gamma > 0$	$\begin{cases} 1, & \ell_i \leq \lambda + \gamma \frac{1}{\sqrt{i} - \sqrt{i-1}} \\ 0, & \text{otherwise} \end{cases}$
Xu <i>et al.</i> [29]	$\sum_{i=1}^n \ln(1 + e^{-\lambda} - v_i)^{(1+e^{-\lambda}-v_i)}$ $+ \ln(v_i)^{v_i} - \lambda v_i, \lambda > 0$	$\frac{1+e^{-\lambda}}{1+e^{\ell_i-\lambda}}$
Zhao <i>et al.</i> [32]	$\sum_{i=1}^n \frac{\lambda \gamma^2}{\lambda v_i + \gamma}, \lambda > 0, \gamma > 0$	$\begin{cases} 1, & \ell_i \leq (\frac{\lambda \gamma}{\lambda + \gamma})^2 \\ 0, & \ell_i \geq \lambda^2 \\ \gamma(\frac{1}{\sqrt{\ell_i}} - \frac{1}{\lambda}), & \text{otherwise} \end{cases}$
Zhang <i>et al.</i> [31]	$-\lambda \sum_{k=1}^K \sum_{i=1}^{n_k} v_i^k - \gamma \sum_{k=1}^K \sqrt{\sum_{i=1}^{n_k} v_i^k},$ $\lambda > 0, \gamma > 0$	$\begin{cases} 1, & \ell_i^k < \lambda + \frac{\gamma}{2\sqrt{i}} \\ \frac{((\frac{\gamma}{2(\ell_i^k - \lambda)})^2 - (i-1))}{m}, & \text{otherwise} \end{cases}$

Table 6 tabulates some examples of self-paced regularizers $g(\lambda, v)$ and their corresponding $v^*(\lambda, \ell)$. We modify their original expressions for better comparison. It is still nontrivial to design self-paced regularizers or analyze their properties according to Definition 3. Besides, though shown to be effective in many applications experimentally, the underlying working mechanism of SPL is still unclear.

One attempt about the underlying working mechanism of SPL is [19]. Starting from SPL regularizers and their minimizer functions, they show that the ASS method used for SPL accords with the *majorization minimization* [26] algorithm implemented on a latent SPL objective, and deduced the latent objective of hard, linear and mixture regularizers. In contrast, we start from a latent loss function $\phi(\lambda, \ell)$ directly and propose self-paced implicit regularizer based on the convex conjugacy theory. We establish the relations between robust loss function $\phi(\lambda, \ell)$, self-paced implicit regularizer $\psi(\lambda, v)$ and minimizer function $\sigma(\lambda, \ell)$. According to Definition 1, $\psi(\lambda, v)$ and $\sigma(\lambda, \ell)$ are derived from latent loss function $\phi(\lambda, \ell)$, thus we can analyze their properties based on the development of $\phi(\lambda, \ell)$ (many loss functions have been widely studied in related areas). We further demonstrate that for SPL with the proposed implicit regularizer, its learning procedure actually associates with certain latent robust loss functions. Thus we can provide some inspirations for the working mechanism of SPL (e.g. its robustness to outliers and heavy noise). Moreover, by establishing the relations between $\phi(\lambda, \ell)$ and $\psi(\lambda, v)$, we can develop new SPL regularizers based on the development of robust loss functions. Specifically, we analyze the relations between self-paced implicit regularizer and HQ optimization. Many robust loss functions and their minimizer functions have been developed and widely used in HQ optimization, and they can be adjusted for self-paced implicit regularizers (some examples are given in Table 1 in main body).

References

- [1] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *ICML*, 2009.
- [2] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011.
- [3] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *JMLR*, 9(Aug):1871–1874, 2008.
- [4] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *TPAMI*, (3):367–383, 1992.
- [5] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *TIP*, 4(7):932–946, 1995.
- [6] R. He, B.-G. Hu, W.-S. Zheng, and Y. Guo. Two-stage sparse representation for robust recognition on large-scale database. In *AAAI*, 2010.
- [7] R. He, T. Tan, and L. Wang. Robust recovery of corrupted low-rankmatrix by implicit regularizers. *TPAMI*, 36(4):770–783, 2014.
- [8] R. He, W. S. Zheng, and B. G. Hu. Maximum correntropy criterion for robust face recognition. *TPAMI*, 33(8):1561–1576, 2011.
- [9] R. He, W.-S. Zheng, T. Tan, and Z. Sun. Half-quadratic-based iterative minimization for robust sparse representation. *TPAMI*, 36(2):261–275, 2014.
- [10] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.
- [11] L. Jiang, D. Meng, T. Mitamura, and A. G. Hauptmann. Easy samples first: Self-paced reranking for zero-example multimedia search. In *MM*, 2014.
- [12] L. Jiang, D. Meng, S.-I. Yu, Z. Lan, S. Shan, and A. Hauptmann. Self-paced learning with diversity. In *NIPS*, 2014.
- [13] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann. Self-paced curriculum learning. In *AAAI*, 2015.
- [14] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *NIPS*, 2010.
- [15] Y. J. Lee and K. Grauman. Learning the easy things first: Self-paced visual category discovery. In *CVPR*, 2011.
- [16] H. Li, M. Gong, D. Meng, and Q. Miao. Multi-objective self-paced learning. In *AAAI*, 2016.
- [17] J. Liang, Z. Li, D. Cao, R. He, and J. Wang. Self-paced cross-modal subspace matching. In *SIGIR*, 2016.
- [18] M. W. Mahoney. Approximate computation and implicit regularization for very large-scale data analysis. In *PODS*, 2012.
- [19] D. Meng and Q. Zhao. What objective does self-paced learning indeed optimize? *arXiv preprint arXiv:1511.06049*, 2015.
- [20] M. Nikolova and R. H. Chan. The equivalence of half-quadratic minimization and the gradient linearization iteration. *TIP*, 16(6):1623–1627, 2007.
- [21] M. Nikolova and M. K. Ng. Analysis of half-quadratic minimization methods for signal and image recovery. *SIAM Journal on Scientific Computing*, 27(3):937–966, 2005.
- [22] L. Orecchia and M. W. Mahoney. Implementing regularization implicitly via approximate eigenvector computation. In *ICML*, 2011.
- [23] R. T. Rockafellar. *Convex analysis*. Princeton university press, 2015.
- [24] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, 2008.
- [25] J. S. Supancic and D. Ramanan. Self-paced learning for long-term tracking. In *CVPR*, 2013.
- [26] F. Vaida. Parameter convergence for em and mm algorithms. *Statistica Sinica*, pages 831–840, 2005.
- [27] N. Wang, T. Yao, J. Wang, and D.-Y. Yeung. A probabilistic approach to robust matrix factorization. In *ECCV*. 2012.
- [28] C. Xu, D. Tao, and C. Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.
- [29] C. Xu, D. Tao, and C. Xu. Multi-view self-paced learning for clustering. In *IJCAI*, 2015.
- [30] X.-T. Yuan and B.-G. Hu. Robust feature extraction via information theoretic learning. In *ICML*, 2009.
- [31] D. Zhang, D. Meng, C. Li, L. Jiang, Q. Zhao, and J. Han. A self-paced multiple-instance learning framework for co-saliency detection. In *ICCV*, 2015.
- [32] Q. Zhao, D. Meng, L. Jiang, Q. Xie, Z. Xu, and A. G. Hauptmann. Self-paced learning for matrix factorization. In *AAAI*, 2015.