

Decomposition-Based Evolutionary Multi-objective Optimization to Self-paced Learning

Maoguo Gong, *Senior Member, IEEE*, Hao Li, Deyu Meng, *Member, IEEE*, Qiguang Miao, *Senior Member, IEEE*, and Jia Liu

Abstract—Self-paced learning (SPL) is a recently proposed paradigm to imitate the learning process of humans/animals. SPL involves easier samples into training at first and then gradually takes more complex ones into consideration. Current SPL regimes incorporate a self-paced regularizer into the learning objective with a gradually increasing pace parameter. Therefore, it is difficult to obtain the solution path of the SPL regime and determine where to optimally stop this increasing process. In this paper, a multi-objective self-paced learning method is proposed to optimize the loss function and the self-paced regularizer simultaneously. A decomposition-based multi-objective particle swarm optimization algorithm is used to simultaneously optimize the two objectives for obtaining the solutions. In the proposed method, a polynomial soft weighting regularizer is proposed to penalize the loss. Theoretical studies are conducted to show that the previous regularizers are roughly particular cases of the proposed polynomial soft weighting regularizer family. Then an implicit decomposition method is proposed to search the solutions with respect to the sample number involved into training. A set of solutions can be obtained by the proposed method and naturally constitute the solution path of the SPL regime. Then a satisfactory solution can be naturally obtained from these solutions by utilizing some effective tools in evolutionary multi-objective optimization. Experiments on matrix factorization and classification problems demonstrate the effectiveness of the proposed technique.

Index Terms—Self-paced learning, machine learning, multi-objective optimization, decomposition.

I. INTRODUCTION

IN machine learning and cognitive science, *self-paced learning* (SPL) [1] proposed by Kumar *et al.* is a recently proposed learning paradigm to avoid a bad local minimum for solving a non-convex optimization task. SPL can trace back to *curriculum learning* (CL) [2] proposed by Bengio *et al.*, in which a curriculum defines a set of training samples organized in ascending order of learning difficulty. Both CL and SPL are inspired by the learning process of humans/animals from the perspective of cognitive science. They involve easier samples into training at first and then gradually take more complex

ones into consideration. Some studies have shown that this learning paradigm is capable of avoiding being trapped in local minima and can obtain better generalization results [3]–[5]. In CL, it is assumed that the curriculum is given in advance and remains unchanged thereafter. However, the SPL regimes can dynamically generate the curriculum based on what the learners have already learned.

In the SPL regimes, a regularization term called self-paced (SP) regularizer is incorporated into the learning objective. A weight variable is used in this regularizer to reflect the easiness of a sample. The SP regularizer specifies the selection of the samples and the calculation of the weights. Then the curriculum and learning model are jointly learned by iteratively updating the weight variable and model parameters. Kumar *et al.* used a binary variable in hard weighting scheme [1] to reflect whether the sample is easy or not. In order to measure the importance of samples, Jiang *et al.* proposed soft weighting methods [6] to assign real-valued weights. Three effective SP regularizers were proposed in [6] to deal with different kinds of data sets. In some cases, the samples can be classified into several groups. For example, in video processing, the samples from the same video can be regarded from the same group. SPL may prefer to select more samples from the same group as they appear to be easy. In order to obtain diverse samples from multiple groups, Jiang *et al.* proposed a new self-paced learning method considering both easiness and diversity, called *self-paced learning with diversity* (SPLD) [7]. SPLD produces a curriculum that reasonably mixes easy samples from several groups, which helps rapidly in obtaining comprehensive knowledge and get better solutions. To deal with prior knowledge, Jiang *et al.* proposed a novel framework called self-paced curriculum learning (SPCL) [8] to combine the merits from both the CL and SPL. SPCL takes both prior knowledge known before training and the learning process during training into consideration. SPL has been successfully applied to various applications, such as segmentation [9], domain adaption [10], dictionary learning [5], long-term tracking [11], reranking [6], action and event detection [7], [8], [12], matrix factorization [12], [13], and co-saliency detection [14].

SPL uses a weight variable to reflect the easiness of a sample and a pace parameter is utilized to control the pace for learning new samples. Note that this parameter gradually increases during iterations. Therefore, a set of solutions can be obtained and these solutions constitute the solution path of the SPL problems. However, there are several issues in the current SPL regimes. On the one hand, it is difficult to set

M. Gong and H. Li are with Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, China (E-mail: omegalihao@gmail.com; gong@ieee.org).

D. Meng is with School of Mathematics and Statistics and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710071, China (E-mail: dymeng@mail.xjtu.edu.cn).

Q. Miao is with School of Computer Science and Technology, Xidian University, Xi'an 710071, China (E-mail: qgmiao@mail.xidian.edu.cn).

This work was supported by the National Natural Science Foundation of China (Grant no. 61772393), the National Program for Support of Top-notch Young Professionals of China, and the National Key Research and Development Program of China (Grant no. 2017YFB0802200).

the gradually increasing pace parameter for getting the entire solution path. In the current SPL regimes, the initial value and the step size of the pace parameter should be given in the initialization. When there is no prior knowledge about the loss, the selection of these parameters is not an easy task and has to be made by experience or by using the trial-and-error method. On the other hand, it is a hard task to determine when to terminate the increasing process in the SPL implementation, i.e., it is difficult to choose a final pace parameter to stop the iterations. SPL tends to obtain a bad solution in the presence of noisy samples when the pace parameter gets larger.

In this paper, we clarify that the current SPL model can be equivalently modeled as a multi-objective model. Then evolutionary multi-objective optimization (EMO) [15]–[17] can be utilized to address these issues existing in the current SPL regimes. Recently, researchers have successfully proved that EMO can achieve the best-so-far theoretically guaranteed approximation performance in some NP-hard problems, such as minimum set cover problem [18], minimum cost coverage problem [19], ensemble pruning [20] and subset selection [21]. Furthermore, particle swarm optimization (PSO) imitates the social behavior of bird flocking and has proved to be an efficient optimization method for solving multi-objective optimization problems [22]–[24]. Multi-objective PSO algorithms optimize multiple conflicting objectives simultaneously to find a set of Pareto optimal solutions. Therefore, the loss term and the SP regularizer term can be selected as two objectives of multi-objective optimization to avoid the selection of the increasing pace parameter. Multi-objective PSO algorithms can obtain a set of solutions, which can be naturally represented as the solution path of the SPL problems. In this way, the current SPL regimes are embedded into this long-term research field for better exploring the SPL insight.

In this study, we propose a novel **multi-objective self-paced learning (MOSPL) model** to address the issues existing in current SPL regimes. The proposed model optimizes two objectives, the loss and the SP regularizer, to find a reasonable compromise between them automatically. However, several weighting regularizers cannot be separated from the learning objectives and cannot be used in the proposed multi-objective model [6], [25]. To address this issue, a polynomial soft weighting regularizer is proposed to penalize the loss. This regularizer defines a soft SP regularizer family. Therefore, it has a more general form and is able to represent more variations. In the implementation of MOSPL, an implicit decomposition method is proposed to search the solutions with respect to the sample number involved into training. Then the entire solution spectrum with respect to the sample number can approximately be obtained by finding the Pareto optimal solutions. In MOSPL, each solution in the solution spectrum has the chance to communicate with its neighboring solutions in the search space. MOSPL gets a set of solutions to constitute the solution path of SPL and obtains more insights into the current SPL regimes. Some off-the-shelf tools in multi-objective optimization can be used to determine the final pace. The previous issues existing in SPL research can be alleviated by the proposed method.

This paper is an extension of work originally reported in the

Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence [12]. The conference paper only considered the linear soft weighting regularizer and did not give the details of the multi-objective optimization algorithm. The contribution of this paper is three-fold: (1) We propose a polynomial soft weighting regularizer, which has a more general form than the previous soft self-paced regularizers. Theoretical studies on the proposed regularizer demonstrate that the linear soft weighting, logarithmic soft weighting and mixture weighting schemes [6], [25] are roughly particular cases of the proposed polynomial weighting regularizer family. (2) An implicit decomposition method is proposed to search the solutions with respect to the sample number involved into training. (3) We substantiate the superiority of MOSPL on matrix factorization, structure from motion, active recognition and multimedia event detection.

This paper is organized as follows. Section II gives the background knowledge on the SPL regime and our motivation of using multi-objective optimization. In Section III, we describe the model and algorithm of MOSPL. Section IV presents the experimental results on matrix factorization and classification problems. Finally, the conclusions are given in Section V.

II. BACKGROUND AND MOTIVATION

A. Self-paced Learning

In this paper, $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ denotes the training dataset, where $\mathbf{x}_i \in \mathbb{R}^m$ is the i th observed sample, and y_i denotes its label. The loss function is represented as $L(y_i, g(\mathbf{x}_i, \mathbf{w}))$ to calculate the cost between the ground truth label y_i and the estimated label $g(\mathbf{x}_i, \mathbf{w})$. \mathbf{w} denotes the model parameters inside the decision function g . SPL uses a weight variable \mathbf{s} to reflect the easiness of a sample. The model parameters \mathbf{w} and the latent weight variable $\mathbf{s} = [s_1, \dots, s_n]$ are jointly learned by minimizing:

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

where $\mathbf{s} \in [0, 1]^n$ and $f(\mathbf{s}; \eta)$ is the SP regularizer. ACS (Alternative Convex Search) is generally used to solve the above equation. 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. The algorithm of SPL is shown in **Algorithm 1**.

As shown in **Algorithm 1**, when \mathbf{w} is fixed, we can obtain the closed-form optimal solutions under different SP regularizers. In the original SPL implementation [1], the weighted training loss and the negative l_1 -norm regularizer $-\|\mathbf{s}\|_1 = -\sum_{i=1}^n s_i$ are minimized with an increasing pace parameter, where s_i is a binary variable. In [6], [13], the researchers proposed the definition of the SP regularizer, which is shown in Appendix A. Then they further extended more efficient formulations of SP regularizers as follows [6], [13].

Linear soft weighting regularizer: This method linearly penalizes the losses, which can be described as the following

Algorithm 1: Algorithm of Self-paced Learning.

Input: The training dataset \mathcal{D} .

Output: Model parameter \mathbf{w} .

```

1 Initialize  $\mathbf{w}^*, \eta$ .
2 while not converged do
3   while not converged do
4     Update  $\mathbf{s}^* = \arg \min_{\mathbf{s}} \mathbb{E}(\mathbf{w}^*, \mathbf{s}; \eta)$ .
5     Update  $\mathbf{w}^* = \arg \min_{\mathbf{w}} \mathbb{E}(\mathbf{w}, \mathbf{s}^*; \eta)$ .
6   end
7   Increase  $\eta$ .
8 end
9 return  $\mathbf{w} = \mathbf{w}^*$ .
```

function:

$$f(\mathbf{s}; \eta) = \eta \left(\frac{1}{2} \|\mathbf{s}\|_2^2 - \sum_{i=1}^n s_i \right), \quad (2)$$

where $\eta > 0$. The closed-form solutions of the SPL model under the linear soft weighting regularizer can be given as follows:

$$s_i^* = \begin{cases} 1 - \frac{L_i}{\eta} & L_i < \eta \\ 0 & L_i \geq \eta. \end{cases} \quad (3)$$

Logarithmic soft weighting regularizer: This scheme is to logarithmically discriminate samples with respect to the losses, which can be written as the following function:

$$f(\mathbf{s}; \eta) = \sum_{i=1}^n \left(\zeta s_i - \frac{\zeta^{s_i}}{\log \zeta} \right), \quad (4)$$

where $\zeta = 1 - \eta$ and $0 < \eta < 1$. Then the closed-form optimal solution for the logarithmic soft weighting is written by:

$$s_i^* = \begin{cases} \frac{1}{\log \zeta} \log(L_i + \zeta) & L_i < \eta \\ 0 & L_i \geq \eta. \end{cases} \quad (5)$$

Mixture weighting regularizer: Mixture method is a hybrid of the “soft” and the “hard” scheme, which can be stated by the following function:

$$f(\mathbf{s}; \eta) = -\zeta \sum_{i=1}^n \log \left(s_i + \frac{1}{\eta_1} \zeta \right), \quad (6)$$

where $\zeta = \frac{\eta_1 \eta_2}{\eta_1 - \eta_2}$ and $\eta_1 > \eta_2 > 0$. Then the closed-form optimal solution is given by:

$$s_i^* = \begin{cases} 1 & L_i \leq \eta_2 \\ 0 & L_i \geq \eta_1 \\ \frac{\zeta}{L_i} - \frac{\zeta}{\eta_1} & \text{otherwise.} \end{cases} \quad (7)$$

When \mathbf{s} is fixed, the existing off-the-shelf learning methods can be employed to obtain the optimal \mathbf{w}^* . The parameter η is the “age” of the SPL model to control the learning pace. In the process of the SPL calculation, we gradually increase η to learn new samples. When η is small, only “easy” samples with small losses will be selected into training. As η grows, more “complex” samples with large losses will be considered. In [25], the authors have proved that the procedure between line 3-6 in **Algorithm 1** can converge to a stationary solution. However, because of the increase of the pace parameter η , the

convergence of the overall algorithm is unclear. There exists several issues in current SPL regimes by introducing a SP regularizer and an increasing pace parameter. Next, we will introduce the motivation of using multi-objective optimization to address these issues.

B. Motivation of Using Multi-objective Optimization

In general, a multi-objective optimization problem (MOP) with m decision variables and l objectives can be stated as

$$\begin{aligned} \min \quad & F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_l(\mathbf{x}))^T \\ \text{s.t.} \quad & \mathbf{x} = [x_1, x_2, \dots, x_m] \in \Omega, \end{aligned} \quad (8)$$

where Ω is the *feasible set*, \mathbf{x} is the *decision vector*, $F : \Omega \rightarrow \mathbb{R}^l$ consists of l real-valued objective functions and \mathbb{R}^l is the *objective space*. The *attainable objective set* is defined as the set $\{F(\mathbf{x}) | \mathbf{x} \in \Omega\}$. In multi-objective optimization, the objective function is a vector, not a scalar value. In the feasible region, usually there is no point that can minimize all objectives simultaneously. Multi-objective optimization [15], [16] is used to obtain a set of Pareto optimal solutions.

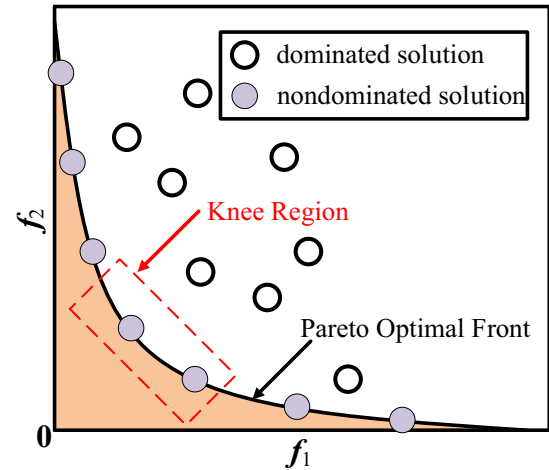


Fig. 1. The schematic diagram of the Pareto front and objective vectors.

Without loss of generality, the multi-objective optimization can be defined for minimization. A decision vector $\mathbf{x}_u \in \Omega$ is said to dominate another vector $\mathbf{x}_v \in \Omega$ if and only if

$$\begin{aligned} \forall i = 1, 2, \dots, n \quad & f_i(\mathbf{x}_u) \leq f_i(\mathbf{x}_v) \\ \wedge \exists j = 1, 2, \dots, n \quad & f_j(\mathbf{x}_u) < f_j(\mathbf{x}_v). \end{aligned} \quad (9)$$

And a solution \mathbf{x}^* in Ω is called a Pareto optimal solution to Eq. (8) if there is no such solution \mathbf{x} in Ω that makes \mathbf{x} dominate \mathbf{x}^* . $F(\mathbf{x}^*)$ is then termed Pareto optimal objective vector. The set of Pareto optimal objective vectors is called the Pareto optimal front (PF) [26]. Fig. 1 shows the Pareto front and objective vectors. The objective vectors on the Pareto front are nondominated. It has been widely investigated that MOPs can be well solved by evolutionary algorithms (EAs). Many multi-objective EAs have been proposed to deal with MOPs for finding a set of Pareto optimal solutions [27], [28]. Moreover, PSO has been used in multi-objective optimization for its simplicity and high speed of convergence [22], [24],

[29]–[32]. As shown in Fig. 1, the knee region is indicated by the dashed box. The solutions in the knee region have the maximum marginal rates of return [33], [34], i.e., the improvement in one objective causes a rapid degradation in other objectives. The solutions in the knee region often have a better trade-off between the two contradictory objectives [34].

In SPL, a pace parameter is introduced to balance the loss and the SP regularizer. SPL increases the pace parameter to gradually involve samples into training. It is difficult to terminate this increasing process and select an appropriate solution. Specifically, in practical scenarios where data contain certain heavy noise/outliers, SPL tends to get a bad solution when the pace parameter gets larger, since those unexpected noise/outliers tend to be involved into training and degenerate the performance of the learning algorithm. In this paper, we clarify that these issues can be alleviated by using multi-objective optimization. Firstly, MOSPL optimizes the loss and the SP regularizer simultaneously to avoid the selection of pace parameters. As described in Section II A, SPL involves easier samples into training at first and then gradually take more complex ones into consideration. The pace parameter controls the pace for learning new samples. In some cases, when we increase the pace parameter, the number of samples remains unchanged. It may waste computational resource. In fact, if we defined the number of samples into training, the pace parameter could be obtained according to this number. Therefore, MOSPL uses the number of samples involved into training to implicitly represent the weight vectors for decomposing the multi-objective optimization problem. MOSPL extends the search range for proper sample selection and makes the SPL process be evaluated at a more global scale. Secondly, MOSPL aims to find the Pareto optimal front, which delivers a complete solution spectrum beyond the traditional SPL, providing only a unique solution under certain pace parameter. Finally, some off-the-shelf tools in multi-objective optimization can be readily used to select a suitable solution from the solution path while it is hard for traditional SPL to easily achieve this task, which generally has to troublesomely employ certain parameter selection strategies or extract prior knowledge.

III. METHODOLOGY

In this section, the proposed MOSPL is described in detail. First, the model of MOSPL is established by simultaneously optimizing the learning objective and the SP regularizer. Then a polynomial soft weighting regularizer is proposed to fit various data with different characteristics, which has a more general form than the previous SP regularizers. Next, an implicit decomposition method based on the pace sequence is proposed to implicitly decompose the MOP into a set of subproblems. Then we present the algorithm of the proposed MOSPL. Finally, the analysis of MOSPL is elaborated on.

A. Multi-objective Self-paced Learning (MOSPL) Model

The general regularization methods get a single solution by finding an optimal parameter to represent the best trade-off between the loss function and the regularizer. In SPL, a

gradually increasing pace parameter is introduced to aggregate the loss term and the SP regularizer term into a scalar objective function. Then a set of solutions can be obtained to represent the solution path of SPL. However, it is difficult to determine this gradually increasing pace parameter in the current SPL regimes. The original SPL implementation uses a greedy strategy to get the solution path and cannot determine where to optimally stop the increasing process. EMO can optimize the conflicting objectives simultaneously and obtain a set of solutions in a single run [35], [36]. It is obvious that EMO can be used to solve the SPL problems for obtaining the entire solution path. In this section, we clarify that the SPL problem can be modeled as a bi-objective optimization problem to alleviate the deficiencies existing in current SPL regimes. The loss term and the SP regularizer term are selected as the two objectives to be optimized simultaneously. The SPL problem in Eq. (1) can be reformulated as the following MOP:

$$\min_{\mathbf{w}, \mathbf{s}} (f_1, f_2)^T = \begin{cases} f_1 : \text{SP regularizer without } \eta, \\ f_2 : \text{The loss function.} \end{cases} \quad (10)$$

where $f_2 = \sum_{i=1}^n s_i L(y_i, g(\mathbf{x}_i, \mathbf{w}))$. Therefore, the new objective function is a vector instead of a scalar objective value. In the above formula, f_1 represents the SP regularizer without the pace parameter. However, as described in Section II, only the hard weighting and the linear soft weighting regularizers can be used in the proposed model because the pace parameter can be separated from these regularizers. Therefore, we propose a polynomial soft weighting regularizer, which defines a soft SP regularizer family and is shown in the next section.

B. Polynomial Soft Weighting Regularizer

As described in Section II, in recent years, several efficient SP regularizers have been proposed, such as hard weighting [1], linear soft weighting, logarithmic soft weighting and mixture weighting [6]. Among them, the logarithmic soft weighting and the mixture weighting regularizers have complex forms, from which the pace parameter cannot be separated. Furthermore, the mixture weighting regularizer introduces another parameter to obtain a larger weight when the loss is of a smaller value and there is no guide for users to select this parameter. Therefore, the logarithmic soft weighting and the mixture weighting regularizers cannot be used in the proposed MOSPL model. The previous regularizers have complex forms and lack uniform formal understanding. In this paper, we propose a novel polynomial soft weighting regularizer to address these issues, which has a simple form and can penalize the loss according to the problem requirements. For example, when the samples have noise, the SP regularizer can penalize the loss to obtain the weights according to the noise level. This scheme is to penalize the loss according to the value of the polynomial order t . Therefore, we can define a soft SP regularizer family, which can be realized by the following set:

$$Q = \left\{ f(\mathbf{s}; \eta) = \eta \left(\frac{1}{t} \sum_{i=1}^n s_i^t - \sum_{i=1}^n s_i \right) \middle| t > 1 \right\}, \quad (11)$$

where $\eta > 0$ and t is the polynomial order.

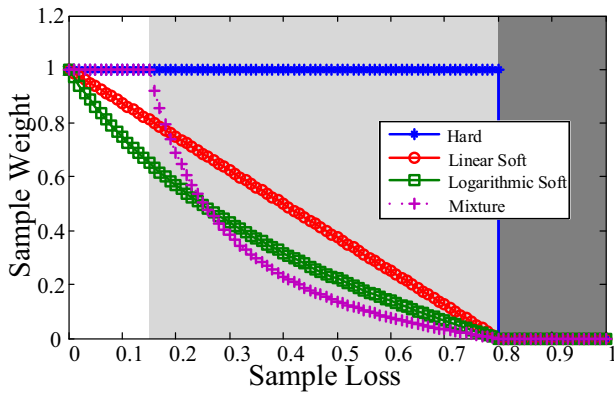


Fig. 2. Comparison of hard, linear soft, logarithmic soft and mixture weighting schemes.

The proposed soft SP regularizer family has a flexible form and can approximate the previous SP regularizers. Furthermore, the proposed regularizer can be easily implemented because each element of Q possesses a closed-form solution. In order to analyze the proposed polynomial weighting regularizer, we deduce the closed-form solutions of the SPL model under this regularizer. Eq. (11) is a convex function of s in $[0,1]$ and thus the global minimum can be obtained at $\nabla_s \mathbb{E}(s) = 0$. Then we have

$$\frac{\partial \mathbb{E}}{\partial s_i} = L_i + \eta(s_i^{t-1} - 1) = 0. \quad (12)$$

The closed-form optimal solution for $s_i (i = 1, 2, \dots, n)$ can be written as:

$$s_i^* = \begin{cases} (1 - \frac{L_i}{\eta})^{1/(t-1)} & L_i < \eta \\ 0 & L_i \geq \eta. \end{cases} \quad (13)$$

TABLE I
THE COMPARISON OF THE HARD WEIGHTING, THE LINEAR SOFT WEIGHTING, THE LOGARITHMIC SOFT WEIGHTING, THE MIXTURE WEIGHTING AND THE PROPOSED POLYNOMIAL SOFT WEIGHTING SCHEMES. IN THESE METHODS, η IS SET TO 0.8. IN THE MIXTURE WEIGHTING SCHEME, $\eta_1 = 0.8$ AND $\eta_2 = 0.15$.

Loss	Hard	Log	$t=1.6$	Linear	$t=2$	Mixture	$t=3$
0.10	1.000	0.749	0.801	0.875	0.875	1.000	0.935
0.15	1.000	0.670	0.726	0.825	0.825	1.000	0.908
0.30	1.000	0.443	0.472	0.638	0.638	0.406	0.798
0.50	1.000	0.231	0.206	0.388	0.388	0.146	0.623
0.75	1.000	0.032	0.010	0.063	0.063	0.016	0.250
0.85	0.000	0.000	0.000	0.000	0.000	0.000	0.000

The proposed polynomial soft weighting scheme conforms to the definitions of SP regularizers [6], [13], which is in Appendix A. Firstly, the polynomial soft weighting regularizer is convex with respect to $s \in [0,1]$. Secondly, $s^*(\eta; L)$ is monotonically decreasing with respect to L , and it holds that $\lim_{L_i \rightarrow 0} s_i^* = 1$, $\lim_{L_i \rightarrow \infty} s_i^* = 0$. It indicates that the SPL model favors easy samples. Finally, s^* is monotonically increasing with respect to η , and it holds that $\lim_{\eta \rightarrow 0} s_i^* = 0$,

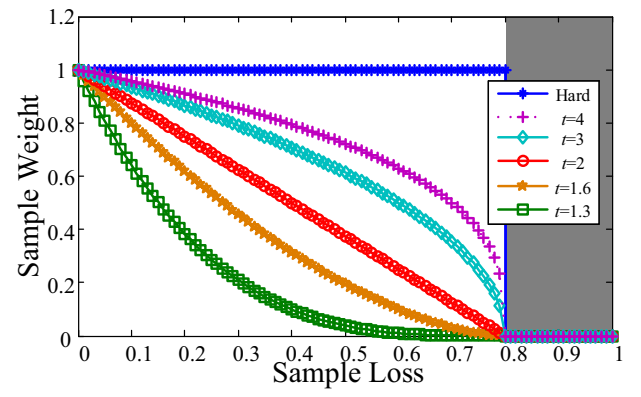


Fig. 3. Comparison of the hard scheme and the proposed polynomial weighting regularizer with polynomial order $t = 1.3, 1.6, 2, 3, 4$.

$\lim_{\eta \rightarrow \infty} s_i^* \leq 1$. When the model “age” gets larger, it tends to incorporate more samples into training.

The closed-form solutions of the SPL model under the other SP regularizers can also be deduced, which have been described in Section II. Fig. 2 shows a comparison of hard weighting, linear soft weighting, logarithmic soft weighting and mixture weighting schemes. Fig. 3 illustrates comparison of hard weighting scheme and the proposed polynomial weighting regularizer. By contrast, when t is set to 2, the polynomial weighting scheme linearly discriminates samples with respect to their losses, which is equivalent to the linear soft weighting method. When t is smaller than 2, the polynomial weighting scheme is similar to the logarithmic soft weighting method. When t is larger than 2, the polynomial weighting scheme obtains higher weight values when the loss is low. Therefore, both the polynomial weighting and mixture weighting schemes can tolerate small errors. The comparison of these weighting schemes is listed in Table I. In a word, the proposed polynomial weighting SP regularizer family can well approximate the linear soft weighting, logarithmic soft weighting and mixture weighting schemes. The polynomial weighting method has a more general form and is able to represent more variations. The users can select a suitable weighting scheme by adjusting the polynomial order.

C. Implicit Decomposition based on the Pace Sequence

MOSPL adopts the decomposition method used in MOEA/D [28]. In this paper, the Tchebycheff technique is used to convert an MOP into a set of scalar objective sub-problems because it is less sensitive to the shape of PF and can be used to find Pareto optimal solutions in both convex and nonconvex PFs. The Tchebycheff approach is defined as

$$g(\mathbf{x} | \boldsymbol{\lambda}^i, \mathbf{z}^*) = \max_{1 \leq j \leq m} \lambda_j^i |f_j(\mathbf{x}) - z_j^*| \quad (14)$$

where $\mathbf{z}^* = (z_1^*, \dots, z_m^*)$ and $z_j^* = \min\{f_j(\mathbf{x}) | \mathbf{x} \in \Omega\}$. \mathbf{z}^* is often unknown before the search. The algorithm uses the lowest f_j -value found during the search to substitute z_j^* [28]. $\boldsymbol{\lambda}^i = (\lambda_1^i, \dots, \lambda_m^i)$ is a weight vector. In general, decomposition-based approaches define a set of evenly distributed vectors and tries to generate solutions that approximate the entire efficient

frontier. However, in the SPL regimes, the computational resource may be wasted when the neighboring subproblems have the same number of selected training samples. In fact, when the number of samples involved into training is known, the pace parameter can be obtained according to this number [6], [7].

In this paper, we design an implicit decomposition method based on the number of samples involved into training. To exploit the preference information, we can define a sample number sequence, called pace sequence for convenience, $\hat{\mathbf{N}} = \{\hat{N}_1, \hat{N}_2, \dots, \hat{N}_{pop}\}$ ($\hat{N}_i < \hat{N}_j$ for $i < j$) representing the number of samples involved into training, where pop is the population size. Each \hat{N}_i indicates the number of samples selected in the i th SPL stage, and $\hat{N}_{pop} = n$ means all samples are involved into training. \hat{N}_i can be considered as an implicit representation of the weight vector λ^i . It is assumed that \hat{N}_i and \hat{N}_j are the number of samples that are involved into training in the i th and j th subproblems, respectively. When $\hat{N}_i < \hat{N}_j$, we have $Lsort_{\hat{N}_i} < Lsort_{\hat{N}_j}$, where $Lsort$ is obtained by sorting the loss L in ascending order. As shown in Eq. (13), we have $\eta_i = Lsort_{\hat{N}_i}$ and $\eta_j = Lsort_{\hat{N}_j}$ to calculate the weights of the samples. Because $Lsort_{\hat{N}_i} < Lsort_{\hat{N}_j}$, the weight of f_1 in the i th subproblem is smaller than that in the j th subproblem. The SPL regimes change the weights of the two objectives to choose different number of samples. Therefore, a relationship can be found between the number of samples and the weights of the two objectives, because the SP regularizer can specify the selection of the samples. We can use the pace sequence to implicitly represent the weight vectors for decomposing the multi-objective optimization problem in Eq. (10).

D. Algorithm of MOSPL

MOSPL aims to find a set of samples that are involved into training. Therefore, the optimization problem in Eq. (10) can be treated as a combinational optimization problem. MOSPL tries to find the optimal subset and then calculates their weight values. Particle swarm optimization (PSO) [37]–[39] is selected as the fundamental optimization tool in the proposed method. PSO imitates the social behavior of bird flocking and works with a population of particles. Each particle has two properties, i.e., its position and velocity. PSO has attracted widespread interests and has been applied to many real world optimization problems [24], [40]. As PSO is simple and effective, applying the PSO mechanism in the decomposition-based multi-objective algorithm may provide a new way for solving the SPL problems. In MOSPL, the offspring should have the same number of nonzero entries as the parents. Fortunately, the discrete PSO algorithm proposed by Zhang *et al.* [41] can well deal with this problem. In [41], the particles can randomly choose directional movement or random movement and the number of nonzero entries in the new position is equal to that in the previous position. In this paper, we extend it to solve the multi-objective problem in Eq. (10). The algorithm of MOSPL is given in **Algorithm 2**. A sequence is used to represent the solution spectrum, which corresponds to a population $P = \{s_1, s_2, \dots, s_{pop}\}$, where s_j is the variable

Algorithm 2: Algorithm of Multi-objective Self-paced Learning.

Input: The position $P = \{s_1, \dots, s_{pop}\}$, the pace sequence: $\hat{\mathbf{N}} = \{\hat{N}_1, \hat{N}_2, \dots, \hat{N}_{pop}\}$, the number of the weight vectors in the neighborhood of each weight vector: T , the max generation $maxgen$, the probability p .

Output: The knee solution.

- 1 **Initialization:** Initialize the position $P = \{s_1, \dots, s_{pop}\}$ constrained with $\|s_j\|_0 = \hat{N}_j, j = 1, 2, \dots, pop$, where $\|s_j\|_0$ counts the number of nonzero entries in s_j . For each particle, work out the T closest particles in Ω based on the Euclidean distance [24].
- 2 Set $gen = 0$;
- 3 **while** $gen < maxgen$ **do**
- 4 **for** $k=1$ **to** pop **do**
- 5 Randomly select one particle from the neighbors as the global best particle;
- 6 **if** $rand(0, 1) > p$ **then**
- 7 Calculate new velocity \mathbf{v}_k by Eq. (15);
- 8 **else**
- 9 Calculate new velocity \mathbf{v}_k by Eq. (16);
- 10 **end**
- 11 Calculate new position s_k by Eq. (17).
- 12 Update the ideal point \mathbf{z}^* ($j = 1, \dots, m$) by

$$\mathbf{z}_j^* = \min\{\mathbf{z}_j^*, f_j(s_k)\}.$$

Update neighborhood solutions [28]: for the i th ($i = 1, 2, \dots, T$) particle in the neighborhood of the k th particle, if $g(s_k | \lambda^i, \mathbf{z}^*) \leq g(s_i | \lambda^i, \mathbf{z}^*)$, then set $s_i = s_k$, $F(s_i) = F(s_k)$.
- 13 **end**
- 14 **end**
- 15 Obtain the solution set with respect to the pace sequence $\hat{\mathbf{N}}$.
- 16 For each solution, four angles $\alpha_1, \alpha_2, \alpha_3$, and α_4 shown in Fig. 4 are computed.
- 17 The largest angle among the four angles is assigned to the solution. Each solution has an angle and the knee point is the solution with the largest angle [34], [42].

to reflect the easiness of j th sample. Obviously, the expected solution spectrum is the population P with the pace sequence $\hat{\mathbf{N}}$. Specifically, for each element s_j along this solution path P , the number of its nonzero entries is \hat{N}_j .

In this paper, $s_k(i)$ and $\mathbf{v}_k(i)$ are used to represent the k th particle's position and velocity. $s_{k,best}(i)$ and $s_{gbest}(i)$ are used to represent the k th particle's self-optimum position and all particles' global optimum position in history. The directional and random movements proposed in [41] are used to update the particles' status. The directional movement is calculated by

$$\mathbf{v}_k(i+1) = T((s_{k,best}(i) - s_k(i)) + (s_{gbest}(i) - s_k(i))), \quad (15)$$

where $s_{k,best}$ represents the personal best position of k th

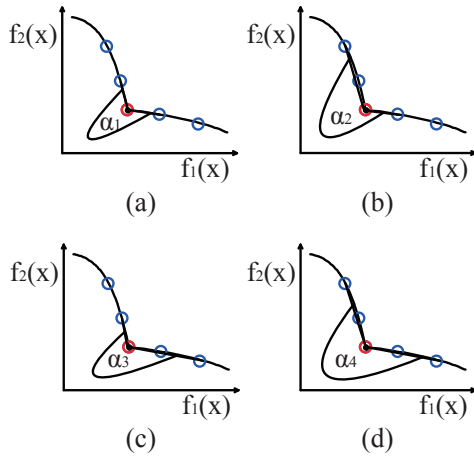


Fig. 4. Four angles in the angle-based method.

particle and s_{gbest} is the global best position. s_{gbest} can be selected from the neighbors to improve diversity [23], [24]. $T(x)$ is the random selection function based on self-experience and social experience. One of x 's positive elements is selected and its value is set to 1. The value of the randomly selected one from x 's negative elements is set to -1. Moreover, the value of the rest of the x 's elements is set to 0. In order to promote diversity, random movement is used in the proposed method, which is described as

$$v_k(i+1) = R(s_k(i)), \quad (16)$$

where $R(x)$ is also a random selection function without considering past experiences. One of the x 's 1 elements is set to -1 and one of the rest is set to 1.

A random selection probability p is used to randomly select directional movement or random movement. After updating the velocity, particles make use of the new velocity to build new position by

$$s_k(i+1) = s_k(i) + v_k(i+1). \quad (17)$$

For each particle in P , we randomly select one particle from its neighbors as the global best particle. Then the velocity is updated by Eq. (15) or Eq. (16) with a random selection probability p . We can update the position by Eq. (17) based on the acquired velocity. Then a local search strategy proposed in [28] is used to update neighborhood solutions. Finally, a solution spectrum with respect to different pace numbers N is expected to be improved in the self-paced learning process.

In this paper, we use the angle-based method [34], [42] for locating the knee on the PF and four points are considered. As shown in Fig. 4, the angle of a solution is determined by its four neighborhood solutions. The core idea of angle-based method is to find the solution with the biggest angle by comparing the biggest angle of every individual between its four angles which is determined by the four neighborhood solutions. The larger the angle of a solution is, the more likely the solution can be considered as the knee.

E. Analysis on MOSPL

Meng *et al.* have proved that the solving strategy on SPL exactly accords with a majorization minimization algorithm implemented on a latent objective and the loss function contained in this latent objective has a similar configuration with non-convex regularized penalty [25]. In SPL, we can obtain the solution s^* as follows:

$$s^*(\eta; L) = \arg \min_{s \in [0,1]} sL + f(s, \eta). \quad (18)$$

We can get the integral of $s^*(\eta; L)$ calculated by Eq. (18) as:

$$F_\eta(L) = \int_0^L s^*(\eta; L) dL + c, \quad (19)$$

where c is a constant.

Now we try to discover more interesting insights under the proposed MOSPL method from this latent SPL objective. To this aim, we first calculate the latent SPL losses under the proposed polynomial soft weighting regularizer (11) by Eq. (19) as follows:

$$F_\eta(L) = \begin{cases} -\frac{(t-1)\eta}{t} (1 - \frac{L}{\eta})^{\frac{t}{t-1}} + c & L < \eta \\ c & L \geq \eta. \end{cases} \quad (20)$$

Then $F_\eta(L)$ with different pace parameters are shown in Fig. 5. Note that when $\eta = \infty$, the latent SPL loss $F_\eta(L)$ will degenerate to the original loss L . There is an evident suppressing effect of $F_\eta(L)$ on large losses as compared with the original loss function L . When L is larger than a certain threshold, $F_\eta(L)$ will become a constant thereafter. This provides a rational explanation on why the SPL regime can perform robust in the presence of extreme outliers or heavy noises: The samples with loss values larger than the age threshold will have no influence on the model training due to their 0 gradients. Corresponding to the original SPL model, these large loss samples will be with 0 importance weights s_i , and thus have no effect on the optimization of model parameters.

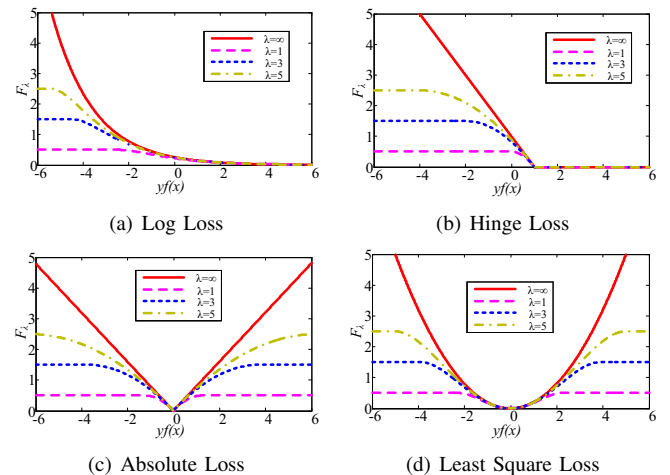


Fig. 5. Graphical illustration for the latent SPL losses conducted by the proposed polynomial soft weighting regularizer on log, hinge, absolute and least square losses. The polynomial order is set to 2 for convenience.

The proposed MOSPL method can alleviate the deficiencies existing in current SPL implementations. The objective function in MOSPL is a vector instead of a scalar objective value. The loss and the SP regularizer are optimized by a decomposition-based multi-objective PSO algorithm to obtain a number of solutions. Compared to the current SPL implementations, each solution in the solution spectrum of MOSPL has the chance to communicate with its neighboring solutions. Furthermore, MOSPL drives the solutions to converge to the Pareto optimal front and obtains the entire solution spectrum with respect to the pace sequence to gain more insights into the SPL problem.

In our last paper, we select the loss term and the linear SP regularizer as the two objectives [12]. However, in recent years, several SP regularizers have been proposed [6]. Among them, the logarithmic soft weighting and the mixture weighting regularizers cannot be used in the proposed multi-objective optimization model. This paper has proposed a polynomial soft weighting regularizer to address this issue. Furthermore, the conference paper in [12] did not give the details of the multi-objective optimization algorithm. In this paper, a multi-objective PSO algorithm is designed as the fundamental tool to optimize the two objectives. In PSO, the directional and random movements are used to update the particles' status. Compared to the conference paper, this paper has conducted many theoretical studies. We have deduced the closed-form optimal solution and calculated the latent SPL losses under the proposed SP regularizer. In the appendix, a proof is given to show that the proposed regularizer can satisfy the three definitions of the self-paced function.

IV. EXPERIMENTAL STUDY

In order to evaluate the effectiveness of MOSPL algorithm, matrix factorization and classification problems are considered in the experiments. We first test the performance of the proposed polynomial soft weighting regularizer. Then the experiments of knee regions and the solution path are exhibited to depict the effectiveness of our method. Finally, four experiments, matrix factorization, structure from motion, active recognition, and multimedia event detection, are conducted to indicate that the proposed method is effective compared with other schemes.

The probability p is used to randomly select directional or random movement. If the value of p is larger, the results will be better and the computational time will be longer [41]. In the experiments, p is set to 0.5. In the proposed method, a pace sequence $\hat{\mathbf{N}}$ is used to control the number of samples involved into training. It is assumed that the pace sequence is an arithmetic progression. The last pace parameter \hat{N}_{pop} is equal to the size of the training dataset. We only need to determine the common difference. Note that the first pace parameter \hat{N}_1 should be greater than zero. Furthermore, the polynomial order t controls the weight values with respect to the loss, which will be discussed thereafter.

A. Experiments on the MOSPL Model

In this section, in order to test the performance of the MOSPL model, matrix factorization problems are considered

in the experiments. The goal of matrix factorization is to factorize an $m \times n$ data matrix \mathbf{Y} , whose entries are denoted as y_{ij} s, into two smaller entries $\mathbf{U} \in R^{m \times r}$ 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} [8], [43]. Matrix factorization have been applied in various disciplines, such as structure from motion [44], face reconstruction and background subtraction [45], [46].

We generated the data as follows: \mathbf{U} and \mathbf{V} represent two matrices and are of size 30×30 . We first randomly generated them with each matrix drawn from the Gaussian distribution $\mathcal{N}(0, 1)$, leading to a ground truth rank-4 matrix $\mathbf{Y}_0 = \mathbf{UV}^T$. Then 40% of the elements were randomly selected and designated as missing entries, 20% of the data were corrupted by uniformly distributed noises over $[-20, 20]$, and the rest data were added to Gaussian noise with $\mathcal{N}(0, 0.01)$.

We considered two widely used loss functions in this paper, which are shown as follows:

The LS loss

$$L(y_{ij}, [\mathbf{UV}^T]_{ij}) = (y_{ij} - [\mathbf{UV}^T]_{ij})^2, \quad (21)$$

and the LAD loss

$$L(y_{ij}, [\mathbf{UV}^T]_{ij}) = |y_{ij} - [\mathbf{UV}^T]_{ij}|. \quad (22)$$

The solver proposed by Zheng *et al.* [47] is modified to calculate the loss values for solving the matrix factorization problem with both the LS and LAD loss. 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 $\hat{\mathbf{U}}$, $\hat{\mathbf{V}}$ represent the outputs from a utilized matrix factorization method.

1) *Test of polynomial soft weighting regularizer*: This section describes several experiments to investigate the performance of the proposed polynomial soft weighting regularizer, which penalizes the loss with a polynomial order t . Graphs and statistical box-plots are given to show the superiority of the proposed SP regularizer.

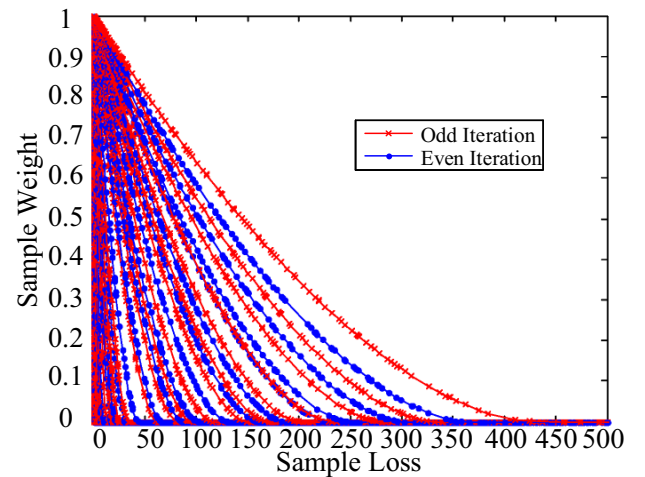


Fig. 6. The weight values obtained by the polynomial soft weighting regularizer with respect to the sample loss at each iteration.

Fig. 6 graphs the weight values at each iteration obtained by the polynomial soft weighting regularizer. In Fig. 6, the

horizontal axis represents the sample loss calculated by the least square method and the vertical axis shows the weight values obtained by the proposed regularizer at $t = 1.6$. The curve shown in the figure moves from left to right to gradually take more samples into consideration. In the early stage, the SPL model only learns from a few samples. Therefore, the weight values of samples with small losses are also set to zero. During iterations, more and more samples are involved into training. Finally, SPL involves a lot of samples into training with different weight values. As shown in Fig. 6, for each iteration, the weight values are obtained by the polynomial soft weighting regularizer with respect to the losses.

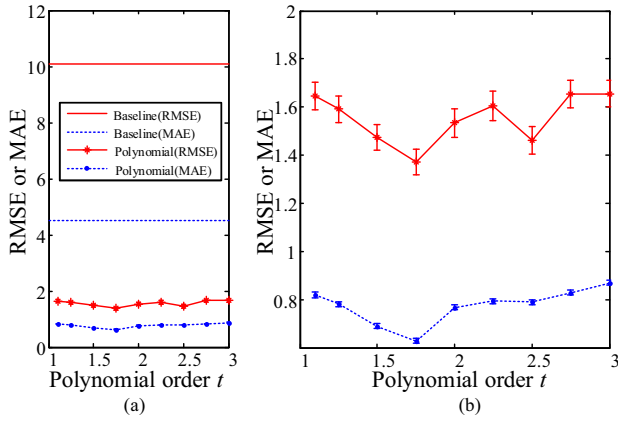


Fig. 7. The values of RMSE and MAE obtained by the polynomial soft weighting regularizer with different polynomial order t . (b) is an enlarged view of (a).

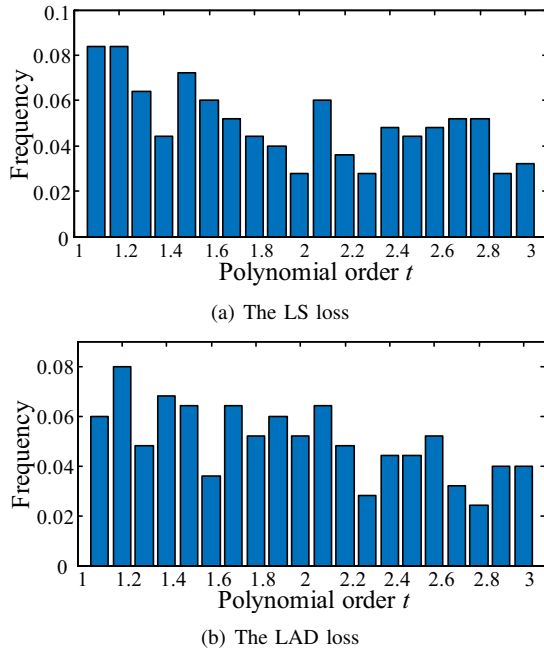


Fig. 8. The frequency of the best t for 250 sets of data. (a) is the results with the LS loss and (b) shows the results with the LAD loss.

Then we test the parameter t of the polynomial soft weighting regularizer, which controls the weight values with respect to the losses. In the experiments, t ranges from 1.1 to 3

($t \neq 1$) and is of some discrete values. The solver proposed in [47] is used as the baseline method, which is modified for calculating the loss values in the SPL and MOSPL models. Fig. 7 shows the mean and standard deviation of the results obtained by the polynomial soft weighting regularizer with different polynomial order t . Fig. 7 (b) is an enlarged view of (a) and the results were obtained from 50 runs to calculate the mean and standard deviation. As shown in Fig. 7 (a), the values of RMSE and MAE obtained by the proposed technique are smaller than those of the baseline method. From Fig. 7 (b), it can be seen that the values of standard deviation are small. Fig. 7 shows that the performance of the proposed polynomial soft weighting regularizer is better than the baseline method. Therefore, the proposed polynomial soft weighting regularizer is not sensitive to the parameter t and can improve the baseline methods by iteratively selecting samples and assigning weights. Next, we randomly generated 250 sets of data and recorded the best parameter t for each set of data. Then we plot the frequency histograms with respect to the polynomial t , which are shown in Fig. 8. We can obtain satisfactory results by turning the parameter t for various datasets. In general, if the data have much noise and many outliers, t should be small to avoid the influence of the samples with large losses.

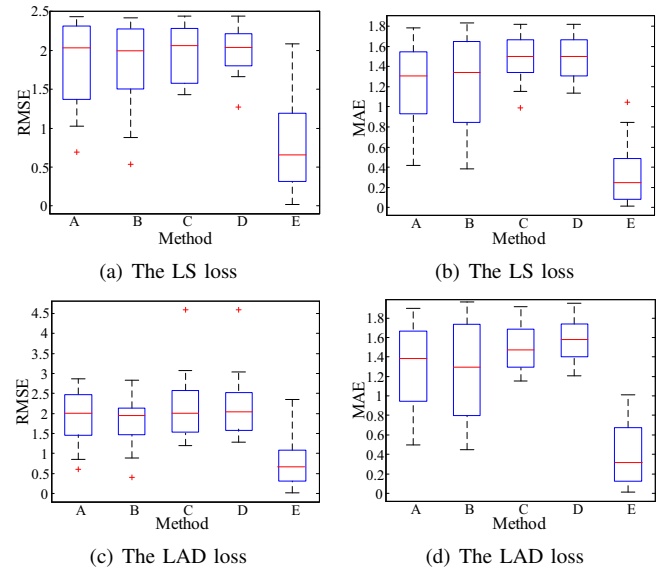


Fig. 9. Box-plots of RMSE and MAE with the LS and LAD loss obtained by A: the hard weighting scheme, B: the linear soft weighting scheme, C: the logarithmic soft weighting scheme, D: the mixture weighting schemes, and E: the proposed polynomial soft weighting scheme.

Then we compared the proposed polynomial soft weighting regularizer with the hard weighting, the linear soft weighting, the logarithmic weighting and the mixture weighting schemes. As stated in Section III, the logarithmic soft weighting and the mixture weighting schemes cannot be used in the proposed MOSPL. Therefore, to get fair results, the original SPL model is utilized in this experiment. The results were obtained from 50 independent runs by using the original SPL model with different SP regularizers. Fig. 9 uses box-plots to show the statistical results obtained by the five weighting schemes. From

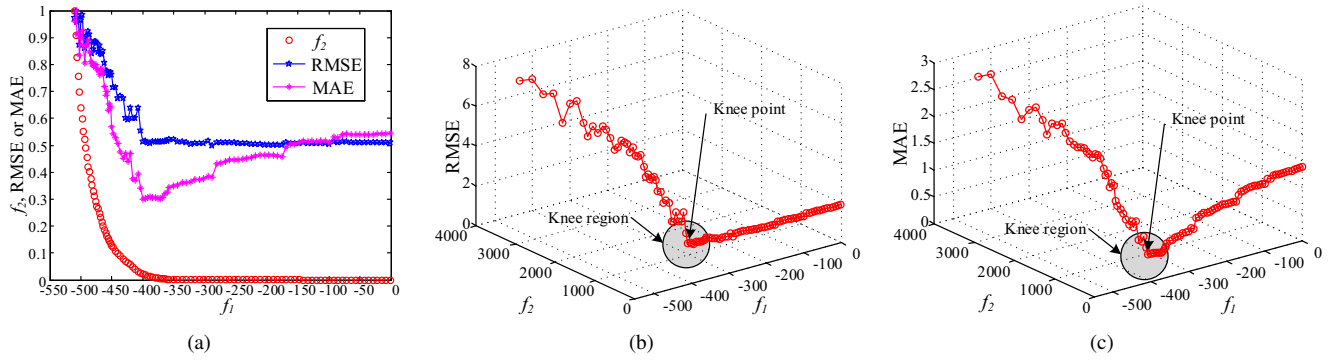


Fig. 10. Relationships among f_1 , f_2 , RMSE and MAE with the LS loss. (a) shows the variation of f_2 , RMSE and MAE with change in f_1 . (B) shows 3-D plot of RMSE, f_1 and f_2 . (c) shows 3-D plot of MAE, f_1 and f_2 .

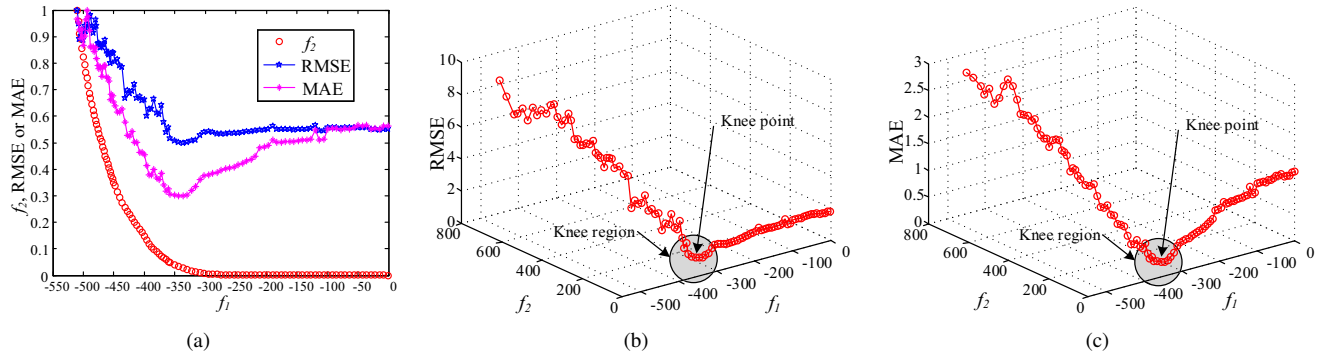


Fig. 11. Relationships among f_1 , f_2 , RMSE and MAE with the LAD loss. (a) shows the variation of f_2 , RMSE and MAE with change in f_1 . (B) shows 3-D plot of RMSE, f_1 and f_2 . (c) shows 3-D plot of MAE, f_1 and f_2 .

Fig. 9, the proposed polynomial soft weighting regularizer gets satisfactory results compared with other schemes. Fig. 2 and Fig. 3 have revealed that the linear soft weighting, the logarithmic soft weighting and the mixture weighting schemes are roughly particular cases of the proposed polynomial soft weighting regularizer family. There is no single weighting scheme that can always work the best for all datasets [6]. The proposed polynomial soft weighting regularizer family can work well on various datasets.

2) *Existence of knee regions in the MOSPL model:* This experiment is conducted to demonstrate the existence of a knee region for this problem. We graph the Pareto front obtained by MOSPL and calculate the values of RMSE and MAE for each solution in the PF.

Fig. 10 and Fig. 11 give the relationship among the SP regularizer (f_1), the learning objective (f_2), RMSE and MAE. The left graphs show 2-D plots of the variance of f_2 , RMSE and MAE with change in f_1 . The middle graphs provide 3-D views of RMSE, f_1 and f_2 . The right graphs depict 3-D views of MAE, f_1 and f_2 . In the left graphs, to better display the relationship among the four factors, the values of f_2 , RMSE and MAE with respect to f_1 are normalized into [0, 1], [0.5, 1] and [0.3, 1], respectively. Fig. 10(a) and Fig. 11(a) show the Pareto fronts obtained by the proposed method. It is easy to distinguish the knee regions in the 3-D plots shown in (b) and (c). Therefore, the knee region does exist on the Pareto front with the LS and LAD loss, which represents good compromise between the learning objective and the SP regularizer. Then

some off-the-shelf tools can be used to find a knee point in the knee region to represent the best trade-off between the two objectives. As shown in Fig. 10(a) and Fig. 11(a), the knee point has small values of RMSE and MAE.

3) *The solution path of the MOSPL model:* Fig. 12 shows the values of objective functions obtained by the SPL and MOSPL models. Since the objective function of the SPL model is a scalar value, the solution path of SPL is obtained by iteratively increasing the pace parameter η . However, the objective function of the MOSPL model is a vector and a set of solutions are obtained for each iteration. From the values of objective functions shown in Fig. 12, the performance of MOSPL is better than that of the SPL model. In Fig. 12, (b) is an enlarged view of (a). In the SPL model, the solution **A** is obtained by learning from the solution **B**. The solution **C** in the MOSPL model can acquire knowledge from its neighbors. The solution path obtained by the MOSPL model seems more meaningful than that of the SPL model because we are able to analysis the knee region in the solution path obtained by the MOSPL model.

To better understand of the convergence of the proposed MOSPL, Fig. 13 shows the tendency curves of RMSE and MAE with respect to iterations. SPL gradually involves more entries into training in the early stage of the SPL implementation. When the iteration continues, SPL cannot terminate this increasing process and tends to get a bad solution. However, the MOSPL model optimizes the learning objective and the SP regularizer simultaneously to get a set of trade-off solutions.

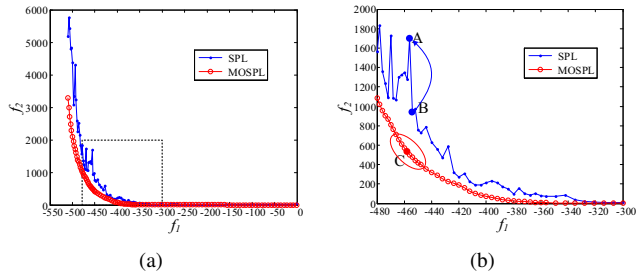


Fig. 12. The values of objective functions obtained by the SPL and MOSPL models. (b) is an enlarged view of (a).

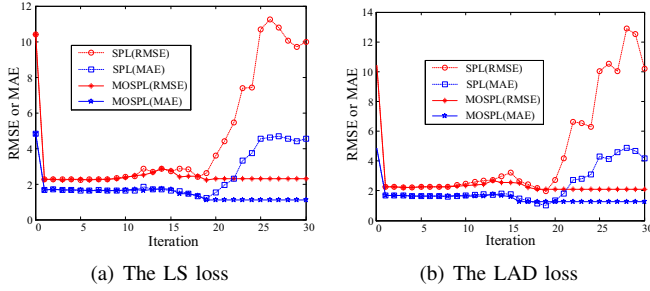


Fig. 13. Tendency curves of RMSE and MAE with respect to iterations.

From Fig. 13, the results obtained by MOSPL are more stable than those obtained by SPL.

B. Comparison of MOSPL Against Other Methods

In this section, we compare the proposed MOSPL model with other algorithms on matrix factorization and classification problems. In matrix factorization, both the synthetic and real data are considered in the experiments, i.e., synthetic matrix factorization and structure from motion. In the experiments of classification, active recognition and multimedia event detection are used to demonstrate the effectiveness of the proposed method.

1) *Synthetic matrix factorization*: In the experiments, four datasets with different noise are used here. Two matrices U and V are of size 30×30 and are randomly generated following Gaussian distribution $\mathcal{N}(0, 1)$. Then 40% of the elements were randomly selected and designated as missing entries, 20% of the data were corrupted by uniformly distributed noises over $[-20, 20]$ or $[-40, 40]$, and the rest data were added to Gaussian noise drawn from $\mathcal{N}(0, 0.01)$ or $\mathcal{N}(0, 0.1)$.

Then we compare the results provided by the proposed technique with those obtained by five other MF methods: WLRA [48], DWiberg [49], RegL1ALM [47], CWM [45], and MoG [46]. The performance of each competing method was evaluated in terms of RMSE and MAE, as the average over the 50 realizations, and reported in Table II. Because SPL tends to get a bad solution with the gradually increasing pace parameter shown in Fig. 13, both the optimal and stable results obtained by SPL are stated in the table. As shown in Table II, both SPL and MOSPL can get satisfactory results compared with other MF algorithms. However, SPL cannot stop at the optimal solution. When the iteration continues, the results of SPL will be bad. The proposed MOSPL can obtain the whole solution path and select the optimal solution. As shown in

Table II, the MOSPL model performs well in the presence of outliers and missing data.


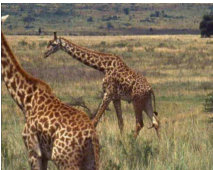
Dataset	<i>Dinosaur</i>	<i>Giraffe</i>
		
Matrix size	72×319	240×166
Rank	4	6
Missing rate	76.92%	30.24%

Fig. 14. Description of SFM data.

2) *Structure from motion*: Structure from motion (SFM) [50] problems can be formulated as MF problems. Therefore, SFM is conducted in this experiment to further test the performance of the proposed MOSPL model. The goal of SFM is to estimate 3-D structure from a sequence of 2-D images which are coupled with local motion information. We considered both rigid and nonrigid SFM problems in the experiments. For rigid SFM, the *Dinosaur* sequence¹ is used to compare with other MF algorithms, which contains 319 feature points tracked over 36 views, corresponding to a matrix Y_0 of size 72×319 with 76.92% missing entries. For nonrigid SFM, the *Giraffe* sequence² is used, which includes 166 feature points tracked over 120 frames. The data matrix Y_0 is of size 240×166 with 30.24% missing entries. The specifications of the two datasets are shown in Fig. 14. In the two datasets, 10% of the entries were randomly chosen and added to outliers, generated from uniform distribution on $[-40, 40]$. The rank was set to 4 and 6 for rigid and nonrigid SFM, respectively [50], [51].

TABLE III
PERFORMANCE COMPARISON OF SEVEN COMPETING MF METHODS ON SFM DATA. THE RESULTS ARE AVERAGED OVER 50 RUNS.

Method	<i>Dinosaur</i>		<i>Giraffe</i>	
	RMSE	MAE	RMSE	MAE
WLRA	5.7838	3.8673	2.8934	1.6932
DWiberg	3.1852	0.9574	1.8783	1.2638
RegL1ALM	2.5794	0.7955	0.6942	0.3723
MoG	3.3523	1.9816	1.4385	1.0463
CWM	10.7489	4.9843	0.8253	0.4011
SPL(Optimal)	1.9759	0.5449	0.6218	0.3091
SPL(Stable)	2.5532	0.8243	0.6749	0.3538
MOSPL	1.8865	0.5032	0.5859	0.2894

The averaged results are obtained by the seven algorithms over 50 runs, which are shown in Table III. It can be seen that both SPL and MOSPL can obtain satisfactory results. However, the stable result of SPL is not good enough because

¹<http://www.robots.ox.ac.uk/~amb/>

²<http://www.robots.ox.ac.uk/~amb/>

TABLE II
PERFORMANCE COMPARISON OF MATRIX FACTORIZATION METHODS IN TERMS OF RMSE AND MAE ON SYNTHETIC DATA. THE RESULTS ARE AVERAGED OVER 50 RUNS.

Criteria		WLRA	DWiberg	RegL1ALM	MoG	CWM	SPL(LS)		SPL(LAD)		MOSPL (LS)	MOSPL (LAD)
							Optimal	Stable	Optimal	Stable		
Uniform Noise: [-20,20] and Gaussian Noise: $\mathcal{N}(0, 0.01)$												
RMSE	Mean	5.6133	5.6431	5.1345	5.6369	5.1348	1.6138	5.0681	1.5727	5.5222	1.5824	1.4633
	Std	0.2117	0.1969	0.3026	0.2256	0.3025	0.5239	0.7156	0.6031	0.7081	0.2853	0.3021
MAE	Mean	3.6325	3.5684	2.8271	3.5558	2.8272	0.8009	3.7249	0.8676	3.6261	0.7751	0.8344
	Std	0.1678	0.2297	0.1910	0.1909	0.1909	0.4252	0.2870	0.5053	0.2737	0.2023	0.1982
Uniform Noise: [-20,20] and Gaussian Noise: $\mathcal{N}(0, 0.1)$												
RMSE	Mean	5.6251	5.6436	5.2149	5.6664	5.2149	1.6476	5.1275	1.8273	5.6208	1.5859	1.7302
	Std	0.1757	0.2065	0.2631	0.1909	0.2631	0.5928	0.9734	0.4592	1.1090	0.1942	0.1834
MAE	Mean	3.6298	3.5014	2.8757	3.5151	2.8759	0.9247	3.7985	1.0383	3.7804	0.8492	0.9552
	Std	0.1914	0.2053	0.2069	0.2116	0.2069	0.4318	0.2626	0.4000	0.2790	0.2134	0.1975
Uniform Noise: [-40,40] and Gaussian Noise: $\mathcal{N}(0, 0.01)$												
RMSE	Mean	11.3334	11.3326	10.1453	11.3362	10.1459	1.5106	11.1684	1.53720	11.0148	1.5049	1.5177
	Std	0.4943	0.4865	0.4562	0.4915	0.4565	0.8158	2.355	0.7559	2.2882	0.4923	0.4562
MAE	Mean	6.9765	6.5010	5.0320	6.4862	5.0325	0.5917	6.7875	0.6136	7.2244	0.4735	0.5266
	Std	0.4724	0.5787	0.3078	0.5345	0.3082	0.4762	0.6162	0.5027	0.6432	0.4598	0.4793
Uniform Noise: [-40,40] and Gaussian Noise: $\mathcal{N}(0, 0.1)$												
RMSE	Mean	11.2182	11.2739	10.1436	11.2780	10.1482	1.2734	11.9830	1.4470	11.0106	1.1095	1.3496
	Std	0.4840	0.4600	0.5321	0.4680	0.5299	0.8951	2.5738	0.8370	2.1421	0.4981	0.4794
MAE	Mean	6.9592	6.5818	5.0585	6.5598	5.0609	0.5363	6.5793	0.6404	6.8739	0.4361	0.5435
	Std	0.3843	0.4431	0.2818	0.4094	0.2811	0.4302	0.6206	0.4500	0.5397	0.3921	0.4200

TABLE IV
PERFORMANCE COMPARISON WITH THE BASELINE METHODS ON HOLLYWOOD2.

Action ID & Name	RandomForest	AdaBoost	BatchTrain	SPL		MOSPL
				Optimal	Stable	
H01: AnswerPhone	20.492	18.350	18.775	33.719	19.583	39.646
H02: DriveCar	68.825	88.729	95.790	95.790	95.790	95.790
H03: Eat	11.041	17.889	71.750	71.750	71.750	71.750
H04: FightPerson	48.067	69.928	81.960	81.960	81.960	81.960
H05: GetOutCar	11.559	28.974	62.787	62.786	62.786	62.786
H06: HandShake	8.206	6.042	42.988	42.982	42.982	42.982
H07: HugPerson	10.846	23.362	16.716	34.378	11.441	37.509
H08: Kiss	40.299	46.424	63.340	60.567	60.539	60.874
H09: Run	46.916	70.865	85.751	79.454	79.276	80.279
H10: SitDown	30.832	66.650	53.595	81.530	81.323	82.055
H11: SitUp	5.307	7.437	35.860	38.870	38.870	38.870
H12: StandUp	35.969	48.997	65.657	80.861	80.841	81.445
MAP($\times 100$)	28.196	41.137	58.164	63.720	60.595	64.662

the SPL model tends to get bad results, for example, the averaged RMSE of *Dinosaur* sequence is increased from 2.5421 to 1.9892. The averaged RMSE and MAE by MOSPL are less than the results obtained by the other algorithms, which substantiates the superiority of the proposed method. The typical recovered tracks of the *Dinosaur* sequence are depicted in Fig. 15 to visualize the results obtained by seven different methods. Obviously, the proposed method can recover the tracks with high quality, which demonstrates the effectiveness of the MOSPL model.

3) *Action recognition*: Action recognition aims to recognize human actions in videos. Hollywood2³ contains 1707 videos generated from 69 different Hollywood movies [52]. The training set has 823 videos and the test set has 884 videos. These videos belong to 12 actions. We extract the improved dense trajectory features from Hollywood2 [53] and apply the spatial and temporal extension on the improved dense trajectory [54]. For more information, please refer to [7]. Mean Average Precision (MAP) is used to evaluate the performance of the competing algorithms [7].

³<http://www.di.ens.fr/~laptev/actions/hollywood2/>

TABLE V
PERFORMANCE COMPARISON WITH THE BASELINE METHODS ON TRECVID MED.

Action ID & Name	RandomForest	AdaBoost	BatchTrain	SPL		MOSPL
				Optimal	Stable	
E006: Birthday Party	1.498	1.696	9.411	11.697	9.670	11.624
E007: Changing a vehicle time	7.819	4.541	19.148	20.826	19.460	26.613
E008: Flash mob gathering	3.357	15.147	23.673	26.207	22.790	29.228
E009: Getting a vehicle unstuck	7.711	10.830	11.349	11.727	10.588	29.923
E010: Grooming an animal	4.065	1.522	5.830	6.930	5.758	9.764
E011: Making a sandwich	2.497	1.076	9.347	12.855	8.893	13.148
E012: Parade	3.767	3.951	16.255	19.519	16.441	19.187
E013: Parkour	7.299	9.012	27.403	27.490	27.490	29.032
E014: Repairing an appliance	3.695	0.581	18.237	17.793	17.793	27.889
E015: Working on a sewing project	0.497	0.342	2.482	2.469	2.169	2.906
E021: Attempting a bike trick	1.076	0.402	3.965	3.809	3.809	7.599
E022: Cleaning an appliance	0.476	0.499	0.932	5.719	0.938	7.319
E023: Dog show	8.696	1.497	6.371	11.999	4.870	18.484
E024: Giving directions to a location	0.205	0.218	0.548	0.615	0.578	3.769
E025: Marriage proposal	0.180	0.127	0.313	0.316	0.316	0.679
E026: Renovating a home	0.188	0.246	1.368	1.483	1.336	4.276
E027: Rock climbing	4.381	3.284	3.288	5.637	2.987	4.266
E028: Town hall meeting	0.572	0.295	2.406	2.857	2.856	6.573
E029: Winning a race without a vehicle	0.650	0.730	1.953	2.679	1.971	14.294
E030: Working on a metal crafts project	2.360	0.410	0.861	0.848	0.848	1.587
MAP($\times 100$)	3.049	2.820	8.257	9.674	8.093	13.408

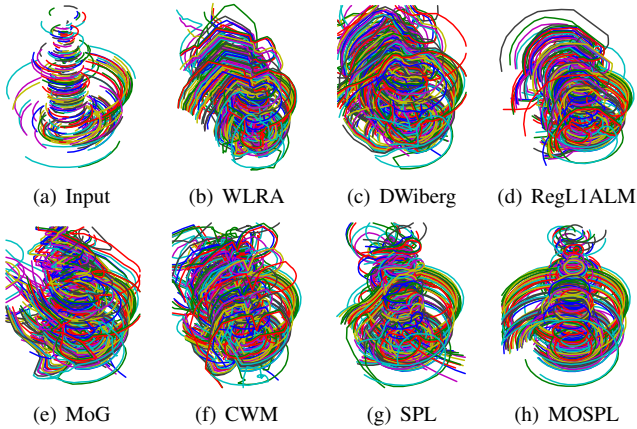


Fig. 15. Recovered tracks from the Dinosaur sequence.

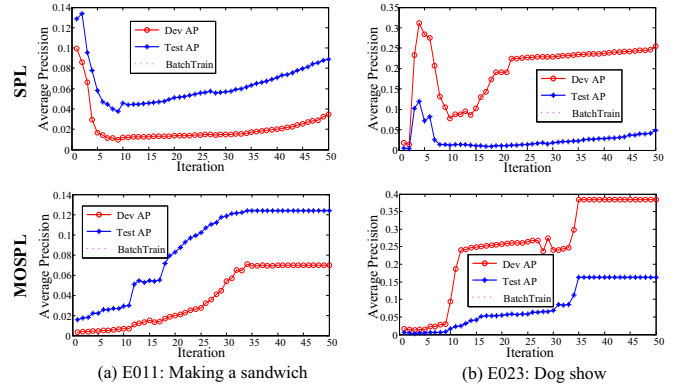


Fig. 17. The validation AP (Dev AP) and the test AP (Test AP) of different iterations on TRECVID MED.

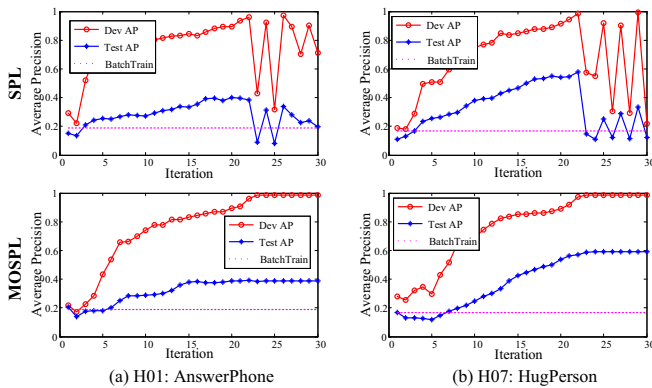


Fig. 16. The validation AP (Dev AP) and the test AP (Test AP) of different iterations on Hollywood2.

Table IV lists the MAP comparison obtained by Random-Forest, AdaBoost, BatchTrain, SPL and MOSPL. BatchTrain

is a standard train approach and simultaneously uses all samples to train a model. Therefore, kernel SVM is selected as the standard train approach [6]–[8]. As shown in Table IV, MOSPL has a better performance than other algorithms. Specially, the stable results of SPL is much worse than the optimal results of SPL because SPL tends to get bad solutions. Fig. 16 shows the validation AP (Dev AP) and the test AP (Test AP) of different iterations on Hollywood2. It is obvious that SPL cannot get stable results after several iterations. The final results of MOSPL are better those of SPL. Therefore, MOSPL is more robust than SPL and can get relatively satisfactory results.

4) *Multimedia event detection*: Multimedia event detection (MED) aims to detect events of interest according to the video content, which is challenging due to occlusions, complex scenes and camera motion [55], [56]. In this experiment,

the TRECVID MED13Test dataset⁴ is used for multimedia event detection. The dataset consists of about 32,000 Internet videos. 3,490 videos from 20 complex events are selected, such as “Birthday party”, “Parade” and so on. The rest are background videos. The official test split released by NIST (National Institute of Standards and Technology) is used. All features are extracted from the video content. The experiment for the TRECVID MED13Test dataset is exhibited as well as the previous experiment.

Table V gives the results obtained by five algorithms. MOSPL has a better performance than other algorithms on 16 out of 20 events. MOSPL achieves a relative improvement of 38.6% compared with the optimal MAP of SPL. Fig. 17 plots the validation and the test AP on two representative events. The results of MOSPL are more stable than those of SPL after several iterations.

V. CONCLUDING REMARKS

SPL uses the greedy strategy to obtain the solution with a gradually increasing pace parameter and it is a hard task to determine where to optimally stop the increasing process of the SPL implementation. Furthermore, SPL tends to obtain a bad solution with a large pace parameter. In this study, a novel multi-objective self-paced learning (MOSPL) algorithm is proposed for SPL calculation. The two objectives, the loss and the SP regularizer, are simultaneously optimized to find a reasonable compromise between them. Since some existing SP regularizers cannot be directly used in the proposed MOSPL model, we have proposed a polynomial soft weighting regularizer to fit various data by turning the polynomial order. Theoretical studies have indicated that the previous regularizers are roughly particular cases of the proposed polynomial soft weighting regularizer family. Then a decomposition-based multi-objective evolutionary algorithm is used as the fundamental optimization tool to optimize the two objectives simultaneously. Experimental studies have demonstrated the effectiveness of the proposed MOSPL algorithm.

The proposed technique brings us a new perspective to the current SPL regimes, and evidently addresses the issues existing in the current SPL regimes. On the one hand, MOSPL enhances the robustness of SPL algorithm to simultaneously optimize the two objectives without the pace parameters. On the other hand, a set of nondominated solutions can be obtained by MOSPL to represent a whole solution path with respect to the pace sequence. We can obtain more insights into SPL calculation based on the entire SPL solution spectrum. In the future, we hope to propose an efficient multi-objective evolutionary algorithm to reduce the time complexity.

APPENDIX A

PROOF: THE POLYNOMIAL SOFT WEIGHTING REGULARIZER IS A SELF-PACED FUNCTION

Definition. (Self-paced function [8]) A self-paced function determines a learning scheme. Suppose that $\mathbf{s} = [s_1, \dots, s_n]^T$ denotes a vector of weight variable for each training sample

and $L = [L_1, \dots, L_n]^T$ are the corresponding loss. η controls the learning pace (or model “age”). $f(\mathbf{s}; \eta)$ is called a self-paced function, if

1. $f(\mathbf{s}; \eta)$ is convex.
2. When all variables are fixed except for s_i , L_i , s_i^* decreases with L_i , and it holds that $\lim_{L_i \rightarrow 0} s_i^* = 1$, $\lim_{L_i \rightarrow \infty} s_i^* = 0$.
3. $\|\mathbf{s}\|_1 = \sum_{i=1}^n s_i$ increases with respect to η , and it holds that $\forall i \in [1, n]$, $\lim_{\eta \rightarrow 0} s_i^* = 0$, $\lim_{\eta \rightarrow \infty} s_i^* \leq 1$.

Theorem. The polynomial soft weighting regularizer is a self-paced function.

Proof. We first prove the proposed polynomial soft weighting regularizer satisfying Condition 1 in Definition. The proposed regularizer is

$$f(\mathbf{s}; \eta) = \eta \left(\frac{1}{t} \sum_{i=1}^n s_i^t - \sum_{i=1}^n s_i \right), \quad (23)$$

and the Hessian matrix \mathbf{H} of f is

$$\begin{bmatrix} \eta(t-1)s_1^{t-2} & 0 & \cdots & 0 \\ 0 & \eta(t-1)s_2^{t-2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \eta(t-1)s_n^{t-2} \end{bmatrix} \quad (24)$$

where $\eta > 0$ and $t > 1$. For the sampled involved into training, s_i is the weight of the i th sample and $s_i \in (0, 1]$. It is obvious that $\eta(t-1)s_n^{t-2}$ is greater than zero. \mathbf{H} is a symmetric diagonal matrix. The eigenvalues of a diagonal matrix are the diagonal elements themselves. Obviously, the main diagonal entries are greater than zero. Then the Hessian of f is positive definite. Therefore, the proposed regularizer is convex.

Then the proposed regularizer satisfying Condition 2 will be proved. Denote $\mathbb{E}_{\mathbf{w}} = \sum_{i=1}^n s_i L_i + f(\mathbf{s}; \eta)$ as the objective with the fixed model parameters \mathbf{w} , where L_i is the loss for the i th sample. The optimal solution \mathbf{s}^* can be written as:

$$s_i^* = \begin{cases} (1 - \frac{L_i}{\eta})^{1/(t-1)} & L_i < \eta \\ 0 & L_i \geq \eta. \end{cases} \quad (25)$$

Obviously, s_i is decreasing with respect to L_i and we have that $\lim_{L_i \rightarrow 0} s_i^* = 1$, $\lim_{L_i \rightarrow \infty} s_i^* = 0$.

Finally, the proposed regularizer satisfying Condition 3 will be proved. Each individual v_i^* increases with respect η in the closed-form solution in Eq. (25). Therefore, $\|\mathbf{s}\|_1 = \sum_{i=1}^n s_i$ also increases with respect to η . In an extreme case, when η approaches positive infinity, we have $\lim_{\eta \rightarrow \infty} s_i^* \leq 1$. Similarly, when η approaches 0, we have $\lim_{\eta \rightarrow 0} s_i^* = 0$.

The proposed polynomial soft weighting regularizer is a self-paced function because it satisfies the three conditions.

REFERENCES

- [1] M. P. Kumar, B. Packer, and D. Koller, “Self-paced learning for latent variable models,” in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2010, pp. 1189–1197.
- [2] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proc. 26th Int. Conf. Mach. Learn.*, Montreal, QC, Canada, 2009, pp. 41–48.

⁴<http://www.nist.gov/itl/iad/mig/med13.cfm>

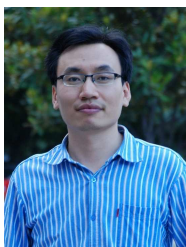
- [3] F. Khan, B. Mutlu, and X. Zhu, "How do humans teach: On curriculum learning and teaching dimension," in *Proc. Adv. Neural Inf. Process. Syst.*, Granada, Spain, 2011, pp. 1449–1457.
- [4] S. Basu and J. Christensen, "Teaching classification boundaries to humans," in *Proc. 27th AAAI Conf. Artif. Intell.*, Bellevue, WA, USA, 2013, pp. 109–115.
- [5] Y. Tang, Y.-B. Yang, and Y. Gao, "Self-paced dictionary learning for image classification," in *Proc. 20th ACM Int. Conf. Multimedia*, New York, NY, USA, 2012, pp. 833–836.
- [6] L. Jiang, D. Meng, T. Mitamura, and A. G. Hauptmann, "Easy samples first: Self-paced reranking for zero-example multimedia search," in *Proc. 22nd ACM Int. Conf. Multimedia*, New York, NY, USA, 2014, pp. 547–556.
- [7] L. Jiang, D. Meng, S.-I. Yu, Z. Lan, S. Shan, and A. Hauptmann, "Self-paced learning with diversity," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 2078–2086.
- [8] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann, "Self-paced curriculum learning," in *Proc. 29th AAAI Conf. Artif. Intell.*, Austin, TX, USA, 2015, pp. 2694–2700.
- [9] M. P. Kumar, H. Turki, D. Preston, and D. Koller, "Learning specific-class segmentation from diverse data," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 1800–1807.
- [10] K. Tang, V. Ramanathan, L. Fei-Fei, and D. Koller, "Shifting weights: Adapting object detectors from image to video," in *Proc. Adv. Neural Inf. Process. Syst.*, South Lake Tahoe, NV, USA, 2012, pp. 638–646.
- [11] J. S. Supančič and D. Ramanan, "Self-paced learning for long-term tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 2379–2386.
- [12] H. Li, M. Gong, D. Meng, and Q. Miao, "Multi-objective self-paced learning," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, 2016, pp. 1802–1808.
- [13] Q. Zhao, D. Meng, L. Jiang, Q. Xie, Z. Xu, and A. G. Hauptmann, "Self-paced learning for matrix factorization," in *Proc. 29th AAAI Conf. Artif. Intell.*, Austin, TX, USA, 2015, pp. 3196–3202.
- [14] 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 *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, 2015, pp. 594–602.
- [15] K. Deb, *Multi-objective optimization using evolutionary algorithms*. New York: Wiley, 2001.
- [16] C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary algorithms for solving multi-objective problems*. Norwell, MA: Kluwer, 2002.
- [17] C. C. Coello, "Evolutionary multi-objective optimization: a historical view of the field," *IEEE computational intelligence magazine*, vol. 1, no. 1, pp. 28–36, 2006.
- [18] Y. Yu, X. Yao, and Z.-H. Zhou, "On the approximation ability of evolutionary optimization with application to minimum set cover," *Artif. Intell.*, vol. 180, pp. 20–33, 2012.
- [19] C. Qian, Y. Yu, and Z.-H. Zhou, "On constrained boolean pareto optimization," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Buenos Aires, Argentina, 2015, pp. 389–395.
- [20] C. Qian, Y. Yu, and Z.-H. Zhou, "Pareto ensemble pruning," in *Proc. 29th AAAI Conf. Artif. Intell.*, Austin, TX, USA, 2015, pp. 2935–2941.
- [21] C. Qian, Y. Yu, and Z.-H. Zhou, "Subset selection by pareto optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 1765–1773.
- [22] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, 2004.
- [23] M. Reyes-Sierra and C. C. Coello, "Multi-objective particle swarm optimizers: A survey of the state-of-the-art," *Int. J. Comput. Intell. Res.*, vol. 2, no. 3, pp. 287–308, 2006.
- [24] M. Gong, Q. Cai, X. Chen, and L. Ma, "Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 82–97, 2014.
- [25] D. Meng, Q. Zhao, and L. Jiang, "A theoretical understanding of self-paced learning," *Inform. Sci.*, vol. 414, pp. 319–328, 2017.
- [26] K. Miettinen, *Nonlinear multiobjective optimization*. Norwell, MA: Kluwer, 1999.
- [27] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.
- [28] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.
- [29] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, 2013.
- [30] W. Hu and G. G. Yen, "Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 1–18, 2015.
- [31] J. Meza, H. Espitia, C. Montenegro, E. Giménez, and R. González-Crespo, "MOVPSO: Vortex multi-objective particle swarm optimization," *Appl. Soft Comput.*, vol. 52, pp. 1042–1057, 2017.
- [32] C. Yue, B. Qu, and J. Liang, "A multi-objective particle swarm optimizer using ring topology for solving multimodal multi-objective problems," *IEEE Trans. Evol. Comput.*, 2017.
- [33] K. Deb and S. Gupta, "Understanding knee points in bicriteria problems and their implications as preferred solution principles," *Eng. Optimiz.*, vol. 43, no. 11, pp. 1175–1204, 2011.
- [34] L. Li, X. Yao, R. Stolkin, M. Gong, and S. He, "An evolutionary multiobjective approach to sparse reconstruction," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 827–845, 2014.
- [35] Y. Matsuyama, "Harmonic competition: a self-organizing multiple criteria optimization," *IEEE Trans. Neural Netw.*, vol. 7, no. 3, pp. 652–668, 1996.
- [36] Y. Jin and B. Sendhoff, "Pareto-based multiobjective machine learning: An overview and case studies," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 3, pp. 397–415, 2008.
- [37] W.-N. Chen, J. Zhang, Y. Lin, N. Chen, Z.-H. Zhan, H. S.-H. Chung, Y. Li, and Y.-H. Shi, "Particle swarm optimization with an aging leader and challengers," *IEEE Trans. Evol. Comput.*, vol. 17, no. 2, pp. 241–258, 2013.
- [38] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, Perth, WA, Australia, 1995, pp. 1942–1948.
- [39] W.-N. Chen, J. Zhang, H. S. Chung, W.-L. Zhong, W.-G. Wu, and Y.-H. Shi, "A novel set-based particle swarm optimization method for discrete optimization problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 2, pp. 278–300, 2010.
- [40] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: A brief survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 2, pp. 262–267, 2011.
- [41] B. Zhang, X. Sun, L. Gao, and L. Yang, "Endmember extraction of hyperspectral remote sensing images based on the discrete particle swarm optimization algorithm," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4173–4176, 2011.
- [42] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multi-objective optimization," in *Proc. 8th Int. Conf. Parallel Problem Solving From Nature*, Birmingham, UK, 2004, pp. 722–731.
- [43] S. P. Chatzis, "Dynamic bayesian probabilistic matrix factorization," in *Proc. 28th AAAI Conf. Artif. Intell.*, Quebec, QC, Canada, 2014, pp. 1731–1737.
- [44] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int. J. Comput. Vision*, vol. 9, no. 2, pp. 137–154, 1992.
- [45] D. Meng, Z. Xu, L. Zhang, and J. Zhao, "A cyclic weighted median method for l_1 low-rank matrix factorization with missing entries," in *Proc. 27th AAAI Conf. Artif. Intell.*, Bellevue, WA, USA, 2013, pp. 704–710.
- [46] D. Meng and F. De la Torre, "Robust matrix factorization with unknown noise," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, 2013, pp. 1337–1344.
- [47] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi, "Practical low-rank matrix approximation under robust l_1 -norm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 1410–1417.
- [48] N. Srebro, T. Jaakkola *et al.*, "Weighted low-rank approximations," in *Proc. 20th Int. Conf. Mach. Learn.*, Washington, DC, USA, 2003, pp. 720–727.
- [49] T. Okatani, T. Yoshida, and K. Deguchi, "Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 842–849.
- [50] A. M. Buchanan and A. W. Fitzgibbon, "Damped newton algorithms for matrix factorization with missing data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, 2005, pp. 316–322.
- [51] Q. Ke and T. Kanade, "Robust l_1 norm factorization in the presence of outliers and missing data by alternative convex programming," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, 2005, pp. 739–746.

- [52] M. Marszałek, I. Laptev, and C. Schmid, "Actions in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, 2009, pp. 2929–2936.
- [53] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, 2007, pp. 1–8.
- [54] Z. Lan, X. Li, and A. G. Hauptmann, "Temporal extension of scale pyramid and spatial pyramid matching for action recognition," *arXiv preprint arXiv:1408.7071*, 2014.
- [55] L. Jiang, A. G. Hauptmann, and G. Xiang, "Leveraging high-level and low-level features for multimedia event detection," in *Proc. 20th ACM Int. Conf. Multimedia*, New York, NY, USA, 2012, pp. 449–458.
- [56] L. Jiang, T. Mitamura, S.-I. Yu, and A. G. Hauptmann, "Zero-example event search using multimodal pseudo relevance feedback," in *Proc. ACM Int. Conf. Multimedia Retrieval*, Glasgow, Scotland, UK, 2014, pp. 297–304.



Qiguang Miao (M'06-SM'17) was born in 1972. He received the Ph.D. degree in computer application technology from Xidian University, Xi'an, China, in 2005.

He is currently a Professor and a Ph.D. Student Supervisor of the School of Computer Science and Technology with Xidian University. His current research interests include machine learning, intelligent image processing, and malware behavior analysis and understanding.



Maoguo Gong (M'07-SM'14) received the B.S. degree in electronic engineering (first class honors) and the Ph.D. degree in electronic science and technology from Xidian University, Xi'an, China, in 2003 and 2009, respectively.

Since 2006, he has been a Teacher with Xidian University. In 2008 and 2010, he was promoted as an Associate Professor and as a Full Professor, respectively, both with exceptive admission. His research interests are in the area of computational intelligence with applications to optimization, learning, data min-

ing and image understanding.

Dr. Gong received the prestigious National Program for the support of Top-Notch Young Professionals from the Central Organization Department of China, the Excellent Young Scientist Foundation from the National Natural Science Foundation of China, and the New Century Excellent Talent in University from the Ministry of Education of China. He is the Vice Chair of the IEEE Computational Intelligence Society Task Force on Memetic Computing, an Executive Committee Member of the Chinese Association for Artificial Intelligence, and a Senior Member of the Chinese Computer Federation. He is also the associate editor of *IEEE Trans. Evolutionary Computation*.



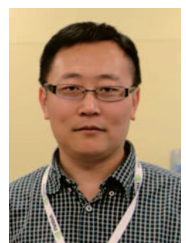
Jia Liu received the B.S. degree in electronic engineering from Xidian University, Xi'an, China, in 2013. He is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems at the School of Electronic Engineering, Xidian University, Xi'an, China.

His research interests include computational intelligence and image understanding.



Hao Li received the B.S. degree in electronic engineering from Xidian University, Xi'an, China, in 2013. He is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems at the School of Electronic Engineering, Xidian University, Xi'an, China.

His research interests include computational intelligence and machine learning.



Deyu Meng (M'13) received the B.Sc., M.Sc., and Ph.D. degrees from Xian Jiaotong University, Xian, China, in 2001, 2004, and 2008, respectively. He was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA, USA, from 2012 to 2014. He is currently a Professor with the Institute for Information and System Sciences, School of Mathematics and Statistics, Xian Jiaotong University. His current research interests include principal component analysis, nonlinear dimensionality reduction, feature extraction and selection, compressed sensing, and

sparse machine learning methods.