IEICE TRANS. INF. & SYST., VOL.E101–D, NO.3 MARCH 2018

LETTER

Self-Paced Learning with Statistics Uncertainty Prior

Lihua GUO[†a)], ***Member***

SUMMARY Self-paced learning (SPL) gradually trains the data from easy to hard, and includes more data into the training process in a self-paced manner. The advantage of SPL is that it has an ability to avoid bad local minima, and the system can improve the generalization performance. However, SPL's system needs an expert to judge the complexity of data at the beginning of training. Generally, this expert does not exist in the beginning, and is learned by gradually training the samples. Based on this consideration, we add an uncertainty of complexity judgment into SPL's system, and propose a self-paced learning with uncertainty prior (SPUP). For efficiently solving our system optimization function, an iterative optimization and statistical simulated annealing method are introduced. The final experimental results indicate that our SPUP has more robustness to the outlier and achieves higher accuracy and less error than SPL.

key words: *self-paced learning, curriculum learning, uncertainty prior, simulated annealing*

1. Introduction

Inspired by the cognitive mechanism and learning process of humans, Kumar et al. tried to fuse curriculum updating in the process of model optimization, and firstly proposed a self-paced learning (SPL) in [1]. The original SPL model gave some weight to all samples according to a specific loss term, and used a general SPL regularization to control the system optimization.

For avoiding bad local minima and improving the generalization performance, many SPL methods have been proposed. Meng et al. further investigated the theoretical insights of SPL in [4]. In SPL, one key issue was to get a better weighting strategy which was determined by a minimization function. Specifically, a definition of self-paced regularization was provided in [2], and three types of self-paced function were proposed including linear soft weighting, logarithmic soft weighting and mixture weighting. In [3], Xu et al. designed a probabilistic smoothed weighting scheme for multi-view clustering, and took into consideration both the complicacy of samples and views. Jiang [8] combined the self-pace learning by the diversity of data distribution to improve the system performance.

In these SPL methods, they always ignore one main aspect, i.e. the self-paced learning method needs to gradually train from easy to hard by a self-pace manner, and this kind of method needs an extra expert to judge the complexity of samples. However, the expert is not available at the beginning. Based on this observation, we assert that the judgment of data complexity is not accurate in the beginning. After gradually training the system, the system gradually becomes smarter, and the judgment of data's complexity will be more accurate. Therefore, self-pace learning with uncertainty prior is proposed in this paper. In our system, an uncertain prior is modeled into the system optimization function, and which will be solved by a statistical method.

Manuscript received July 27, 2017.
Manuscript revised November 3, 2017.
Manuscript publicized December 13, 2017.

[†]The author is with School of Electronic and Information Engineering, South China University of Technology, Guangzhou, 510641, China.
a) E-mail: guolihua@scut.edu.cn

DOI: 10.1587/transinf.2017EDL8169

2. Review of Self-Paced Learning

Given a training dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$ with n samples, where $x_i \in R^d$ 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 x_i in classification model). $f(., W)$ denotes the learned model and $W = [w_1, w_2, \ldots, w_n]$ is the model's parameter, $L(y_i, f(x_i, W))$ is the function which calculates the loss of i-th sample. SPL aims to optimize the model from easy to hard samples gradually in a self-paced manner. The objective of SPL is to jointly optimize the model parameter w and the latent sample weights $V = [v_1, v_2, \ldots, v_n]$ via a following minimization problem:

$$\min_{W,V} E(W, V; \lambda) = \sum_{i=1} v_i L(y_i, f(x_i, W)) + g(\lambda, v_i) \quad (1)$$

where g(.) is called the self-paced regularization, and λ is the penalty parameter that controls the learning pace. Alternative convex search (ACS) is generally used for Eq. (1), which alternatively optimizes W and V while fixing the other. Specifically, given sample's weights V, it is a weighted loss minimization problem that is independent of function g(.). Moreover, given model parameters W, the optimal weight of i-th sample can be obtained via

$$\min_{v_i} v_i L(y_i, f(x_i, W)) + g(\lambda, v_i) \quad (2)$$

since $\ell_i = L(y_i, f(x_i, W))$ is a constant when W is given, the optimum value of v_i is uniquely determined by the corresponding minimum function $v_i \ell_i + g(\lambda, v_i)$. By gradually increasing the value λ, more hard samples are included into the training process in a self-paced manner. The parameter λ is not static, and is increased for developing the system maturation. More details of SPL are in [4].

3. Self-Paced Learning with Uncertainty Prior

Self-paced learning (SPL) gradually includes more data into the training process in a self-paced manner after increasing the penalty of SPL regularization during optimization. When implementing this self-paced learning, the first task is to classify the complexity level of different samples, and then the sample can be gradually trained according to their complexities. Therefore, it needs an expert to make this judgment. However, at the beginning, the system is very young without a discriminative ability, and progressively train to become the wiser. On the contrary, the system must be smarter to judge the complexity of samples. It is contradicting, and seems like a chicken and egg problem. For solving this problem, we give an uncertainty prior to the complexity level of samples. This uncertainty prior meets two criteria as follows,

1) The uncertainty is much when the system is young. On the contrary, the uncertainty must be little when the model becomes mature.

2) The uncertainty prior can be easily combined into the self-pace learning.

Assuming $V = [v_1, v_2, \ldots, v_n]$ is the weight that is learned by the SPL method at a certain parameter λ, we model the true weight of sample $\hat{V} = [\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_n]$ as a stochastic variable following a Gaussian distribution when giving the weight V, which is as follows,

$$P\left(\hat{V} \mid V\right) = N(V, \lambda^{-1}) \tag{3}$$

where λ is the parameter that controls the learning pace. The mean of Gaussian distribution is the weight V, and its deviation is the reciprocal of parameter λ. When system is young, the value of parameter λ is small. This Gaussian distribution has a big variance, it means the variable $\hat{V}$ has a big deviation with the weight V, which indicates the weight V is not certain to match the complexity of samples. Otherwise, the variable $\hat{V}$ has a small deviation with the weight V, which indicates the weight V is certain to match the complexity of samples. This character accords with the first criterion of uncertainty prior. In our method, we should firstly calculate the weight V using the optimization method in SPL at a certain parameter λ, and then sample the weight $\hat{V}$. The loss value of system with the weight $\hat{V}$ can be calculated as follows,

$$E(W, \hat{V}; \lambda) = \sum \hat{v}_i L\left(y_i, f\left(x_i, W\right)\right) + g\left(\lambda, \hat{v}_i\right) \tag{4}$$

The sampling weight $\hat{V}$ is accepted with a certain probability according to the relationship of loss value. The final goal is that the relationship of loss value with these two weights V and $\hat{V}$ must meet the following criterion,

$$E(W, \hat{V}; \lambda) < E\left(W, V; \lambda\right) \tag{5}$$

which means that the system with the weight $\hat{V}$ has more generalization than that with the weight V, i.e. the loss of system with the weight $\hat{V}$ is less than that with the weight V. Finally, our system gradually increases the value λ for developing the system maturation, and repeats the loop of weight calculation until the learning pace parameter λ reaches one.

Our processing is based on the common framework of the self-paced learning. If the loss function is the mean square loss, our system is the regression with self-paced regularization. If the loss function is the hinge loss, our system is the classification with self-pace regularization. Since $\hat{V}$ is a stochastic variable, and could not be explicitly calculated, we sample the value around the weight V using a Gaussian distribution, and use an approximating inference, i.e. simulate annealing, to estimate the true weight of sample $\hat{V}$. Simulated annealing (SA) is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a heuristic to approximate the global optimization in a large search space. Before the simulate annealing, we should calculate the solution V in SPL at a certain parameter λ. Let us use the mean square loss with a soft linear weighting as an example, which is as follows,

$$\min_{W,V} E = \sum v_i \|y_i - w_i x_i\|^2 + \frac{1}{\lambda}\left(\frac{1}{2}\|v\|_2^2 - \sum_{i=1}^{n} v_i\right) \tag{6}$$

where it is a standard SPL optimization, and can be solved using ACS as follows,

a) Fix V, and solve W, it will be simplified as follows,

$$\min_{W} \sum v_i \|y_i - w_i x_i\|^2 \tag{7}$$

which is a traditional MSE regression method, and can be solved by a gradient-based method or coordinate-based method.

b) Fix W, and solve V, it will be simplified as follows,

$$\min_{v} \sum v_i \ell_i + \frac{1}{\lambda}\left(\frac{1}{2}\|v\|_2^2 - \sum_{i=1}^{n} v_i\right), \ \ell_i = \|y_i - w_i x_i\|^2 \tag{8}$$

which can be solved as below,

$$v_i = \begin{cases} 1, & \ell_i < \lambda \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

We sample the value $\hat{V}$ around the weight V using a Gaussian distribution, and use an approximating inference, i.e. simulate annealing, to estimate the weight of sample $\hat{V}$. The pseudo code of our method is shown in Table 1.

4. Experiment and Discussion

In our SPUP method, two parameters need to be tuned, i.e. the learning pace update ratio θ and the rate of temperature descent α. In the simulate annealing, the rate of temperature descent α always sets at 0.99 for slowly decreasing the temperature in order to achieve the more perfect solution. Therefore, our system also sets the rate of temperature descent α as 0.99. Next, we choose a real video dataset, i.e. Hollywood2, to evaluate the system performance when using different pace update ratios. Hollywood2 was collected

Table 1 Our SPUP algorithm's pseudo code

Input: Dataset $D = \{x_i, y_i\}$, the initial regularization parameter λ, and the learning pace update ratio θ

Repeat {main Loop}

Fix V, and solve W, the optimum mapping matrix W can be solved by gradient-based methods or coordinate-based methods.

Fix W, and solve V, we calculate the loss of each sample, and assign the weight v_i to each sample according to Equ.11

Simulate annealing,

a) Set initial temperature T_t, $t = 0$, and rate of temperature descent α;

b) Calculate V using Equ.11 at a certain parameter λ;

c) Calculate E_{old} in Equ.8 using V and W;

d) Sample the value $\hat{V}$ around V using a Gaussian distribution $N(V, \lambda^{-1})$;

e) Calculate E_{new} in the system optimization function (Equ.6) using the weight $\hat{V}$ and the mapping matrix W;

f) If $E_{new} < E_{old}$, the new weight $\hat{V}$ will be accepted, otherwise, only accepted randomly with probability $e^{-\frac{E_{new}-E_{old}}{T}}$;

g) Update the weight $V \leftarrow \hat{V}$, $E_{old} \leftarrow E_{new}$, and the temperature $T_{t+1} = \alpha T_t$;

h) Until $E_{new} - E_{old} < \varepsilon$, or the temperature $T < \varepsilon$, otherwise go step (d).

Update the learning pace parameter $\boldsymbol{\lambda}(\lambda = \theta\lambda)$, until the learning pace parameter λ reaches one.

from 69 different Holly-wood movies [6]. It contains 1,707 videos belonging to 12 actions. The improved dense trajectory feature is extracted and further represented by the Fisher vector. The recognition accuracy of the system keeps stable, which means the learning pace update ratio is not the critical parameter for the final system performance. In our method, we set the learning pace update ratio as 1.03.

4.1 Robustness of Noise and Outlier

A toy dataset with 2 dimensions data is created, and all data are plotted as 2-dimension points as shown in Fig. 1 (a). Firstly, the SPL and our SPUP method should select some random samples as the initial training set, as shown in Fig. 1 (b). Based on these selected samples, the SPL and our SPUP method train an initial model, and use this model to make a judgment about the complexity of all samples. Secondly, the SPL and our SPUP will choose these easy samples as shown in Fig. 1 (c) and Fig. 1 (e) respectively. Thirdly, when the system becomes ever more mature, the SPL and SPUP system will choose more samples to train the model. When the initial classifier is distorted by noise or some outliers, the system will give wrong judgment because the SPL method uses the initial classifier as an expert, which is illustrated in Fig. 1 (d). In our SPUP, we give an uncertainty prior to the weight of initial training data, therefore, initial classifier is not critical during judging the complexity of samples, and the system can gradually adapt the noise and outlier to obtain more reasonable classifier, which

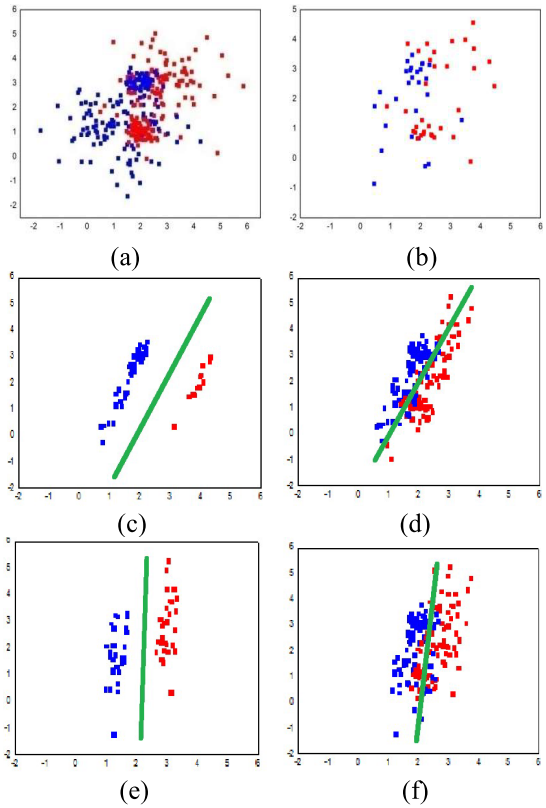

Fig. 1 2-Dimension distribution of a toy dataset and the selection of training data. (a) the original data; (b) the initial selection; (c–d) the selection with the increase of system maturation using SPL method; (e–f) the selection with the increase of system maturation using our SPUP method.

is shown in Fig. 1 (f). This experiment result indicates that SPUP has more robustness to the noise and outlier than the SPL method.

4.2 Experiment on Three Video Datasets

Our comparison baseline methods include: 1) Random Forest [7] is a robust bootstrap method that trains multiple decision trees using randomly selected samples and features. 2) AdaBoost [7] is a classical ensemble approach that combines the sequentially trained "base" classifiers in a weighted fashion. Samples that are misclassified by one base classifier are given greater weight when used to train the next classifier in sequence. 3) Batch-Train represents a standard training approach in which a model is trained simultaneously using all samples; 4) SPL is a method that trains models gradually from easy to more complex samples [1]; 5) SPLD is an improved SPL method by fusing the ease and diverse samples into a general regularizer [8]. Following [8], our method also formalizes the preferences for both easy and diverse samples into a general regularizer, and further samples some values around the sample's weight. At each iteration, our method uses the simulate annealing to accept or reject the training data instead directly selecting data by judging their loss value.

Three representative video datasets are selected to val-

Table 2 The overall MAP comparison of three video datasets.

Datasets	MED	HollyWood2	Olympic Sports
Random Forest [7]	3.0	28.2	63.32
AdaBoost [7]	2.8	41.14	69.25
BatchTrain	8.3	58.16	90.61
SPL [1]	8.6	63.72	90.83
SPLD [8]	9.8	66.65	93.11
Our SPUP	14.65	80.67	98.35

idate our method effectiveness, and the performance is evaluated using MAP to validate the efficiency of all methods:

Multimedia event detection dataset (MED). This data is a collection of videos from TRECVID MED13Test, which consists of about 32,000 Internet videos. This is a total of 3,490 videos from 20 complex events, and the rest are background videos. For each event 10 positive examples are given to train a detector, which is tested on about 25,000 videos. The official test split released by NIST (national Institute of Standards and Technology) is used [9]. The goal of MED is to detect events of interest, e.g. "Birthday Party" and "parade", only based on the video content. The task is very challenging due to complex scenes, camera motion, occlusions, etc. [10]. A deep convolutional neural network is trained on 1.2 million ImageNet challenge images from 1,000 classes to represent each video as a 1,000-dimensional vector. Following [6], the performance is evaluated using MAP (Mean Average Precision).

Hollywood2 is collected from 69 different Holly-wood movies [6]. It contains 1,707 videos belonging to 12 actions, splitting into a training set (823 videos) and a test set (884 videos). The improved dense trajectory feature is extracted and further represented by the Fisher vector.

Olympic Sports dataset consists of athletes practicing different sports collected from YouTube. There are 16 sport actions from 783 clips. We use 649 for training and 134 for testing as recommended in [5].

Table 2 lists the overall MAP comparison. It is worth emphasizing that MED dataset is a very challenging problem, and the MAP of all methods are little low, and our method achieves 49.5% relative improvement over SPLD is a notable gain. Random Forest and AdaBoost yield poorer performance. This observation is in agreement with the study in literature [9] that SVM is more robust on event detection. In the HollyWood2 and Olympic Sports datasets, our SPUP achieve 21% and 5.6% improvement respectively.

4.3 System Complexity and Time Cost Analysis

In every iteration, our SPUP method first calculates the mapping matrix W and the weight V like SPL method, then samples some values, and finds the optimal value using the simulated annealing. Therefore, time costing of our SPUP method will be more than the SPL method. We use the MED, HollyWood2 and Olympic Sports datasets as the evaluated datasets, which are evaluated with a PC machine running in an Intel i5-2400, 3.10 GHz CPU with 8.00 GB RAM. The evaluation metrics include two aspects: training time and testing time. We compare the time cost during training and testing, which are shown in Table 3. When comparing the time cost of training, our SPUP has the largest time cost among three methods. It is because that our method needs an extra operation, i.e. simulated annealing, to search the best optimal value. However, when comparing the time cost of testing, the time cost of three methods is same. It is because that the discriminative model is same, and the only difference is that they have different mapping matrix W. In the real application, the system always trains the model using an off-line mode, the training time cost is not critical importance, and some paralleled techniques, e.g. distributed computation and cloud computation, can be used to improve the speed of training procedure. Therefore, our SPUP is practically useful.

Table 3 The time cost when evaluating three video datasets.

The time cost of training (Minute)			
	SPL [1]	SPLD [8]	Our SPUP
MED	58.43	69.54	195.32
HollyWood2	10.65	12.44	45.68
Olympic Sports	8.31	11.21	39.73
The time cost of training (Second)			
	SPL [1]	SPLD [8]	Our SPUP
MED	0.79	0.79	0.79
HollyWood2	0.38	0.38	0.38
Olympic Sports	0.13	0.13	0.13

5. Conclusion and Discussion

Since the uncertainty is involved in the self-paced learning (SPL) at the beginning of training, we embed an uncertainty prior into the optimization function of SPL, and propose a self-paced learning with uncertainty prior (SPUP). SPUP considers the true weight of samples as a stochastic variable with a Gaussian distribution. For solving this minimization function, an iterative optimization and statistical simulated annealing method are used. The final experimental results indicate that our method has more robustness to the noise and outlier than the SPL. However, our system needs more iteration time. The computation cost of our system is the major disadvantage when comparing with the SPL. In future, it needs further research how to decrease the computation cost of training.

Acknowledgements

This work was partially supported by Natural Science Foundation of China (61673182), Natural Science Foundation of Guangdong Province (2015A030313210), GuangZhou Science and Technology Program (201707010141) and the Fundamental Research Funds for the Central Universities (2015ZM138).

References

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

[2] L. Jiang, D. Meng, T. Mitamura, and A.G. Hauptmann, "Easy samples first: Self-paced reranking for zero-example multimedia search," Proceedings of the 22nd ACM international conference on Multimedia, ACM, pp.547–556, 2014.

[3] C. Xu, D. Tao, and C. Xu, "Multi-view self-paced learning for clustering," Proceedings of the 24th International Conference on Artificial Intelligence, 2015.

[4] D. Meng, Q. Zhao, and L. Jiang, "A theoretical understanding of self-paced learning," Information Sciences, vol.414, pp.319–328, 2017.

[5] J.C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," European conference on computer vision, Springer Berlin Heidelberg, 2010.

[6] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," Computer Vision and Pattern Recognition, 2009, CVPR 2009, IEEE Conference on, IEEE, 2009.

[7] L. Breiman, "Using Iterated Bagging to Debias Regressions," Machine Learning, vol.45, no.3 pp.261–277, 2001.

[8] L. Jiang, et al., "Self-paced learning with diversity," Advances in Neural Information Processing Systems, 2014.

[9] P. Over, et al., "Trecvid 2014–an overview of the goals, tasks, data, evaluation mechanisms and metrics," Proceedings of TRECVID, 2014.

[10] L. Jiang, T. Mitamura, S.-I. Yu, and A.G. Hauptmann, "Zero-example event search using multimodal pseudo relevance feedback," Proceedings of International Conference on Multimedia Retrieval, ACM, pp.297–304, 2014.