Robust Softmax Regression for Multi-class Classification with Self-Paced Learning

Yazhou Ren[1,*], **Peng Zhao**[1], **Yongpan Sheng**[1], **Dezhong Yao**[2], and **Zenglin Xu**[1,*]
[1]SMILE Lab & Big Data Research Center, School of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu, 611731, China.
[2]Key Laboratory for NeuroInformation of Ministry of Education, School of Life Science and Technology,
University of Electronic Science and Technology of China, Chengdu, 611731, China.
yazhou.ren@uestc.edu.cn, zlxu@uestc.edu.cn

Abstract

Softmax regression, a generalization of Logistic regression (LR) in the setting of multi-class classification, has been widely used in many machine learning applications. However, the performance of softmax regression is extremely sensitive to the presence of noisy data and outliers. To address this issue, we propose a model of robust softmax regression (RoSR) originated from the self-paced learning (SPL) paradigm for multi-class classification. Concretely, RoSR equipped with the soft weighting scheme is able to evaluate the importance of each data instance. Then, data instances participate in the classification problem according to their weights. In this way, the influence of noisy data and outliers (which are typically with small weights) can be significantly reduced. However, standard SPL may suffer from the imbalanced class influence problem, where some classes may have little influence in the training process if their instances are not sensitive to the loss. To alleviate this problem, we design two novel soft weighting schemes that assign weights and select instances locally for each class. Experimental results demonstrate the effectiveness of the proposed methods.

1 Introduction

Classification typically involves a supervised learning scenario which studies a model based on the training data samples with known labels and then make predictions on new data (test data). In the past couple of decades, a large number of classification models have been proposed, such as support vector machines (SVM) [Hsu and Lin, 2002], decision tree [Krawczyk *et al.*, 2014], naive Bayes [Farid *et al.*, 2014], neural network [Anand *et al.*, 1993], and so on. Even so, logistic regression (LR) is one of the most widely used methods for binary classification task due to its ease of understanding and implementation, good performance, and high efficiency [Freedman, 2009; Cox, 1958; Walker and Duncan, 1967]. For multi-class classification, LR usually applies the one-vs-all strategy that trains K (the number of classes) binary classification models, each treating the instances in one class as positive and all the other instances as negative. Each model provides a predicted probability of a test data instance and this instance will be assigned to the class that gives the largest probability.

By contract, softmax regression (SR) [Bishop, 2006; Bhning, 1992], also known as multi-class logistic regression, can also be used to solve multi-class classification tasks. In multi-class classification, the classes are usually mutually exclusive. Thus, it is better to use SR instead of building K separate LR models. SR is also less time consuming than LR because SR only needs to train the classification model once. SR has also been widely used in many machine learning applications. However, noisy data and outliers can significantly affect the performances of both LR and SR. To enhance the effectiveness of LR, several robust methods of LR have been proposed recently [Shafieezadeh-Abadeh *et al.*, 2015; Tibshirani and Manning, 2014; Feng *et al.*, 2014]. However, as far as we know, none of work has been done to improve the performance of SR for multi-class classification from the perspective of self-paced learning.

Recently, a new machine learning framework named self-paced learning (SPL) [Kumar *et al.*, 2010] has attracted a lot of attentions. The idea comes from the fact that people usually learn better when they start with simple knowledge and then gradually consider complex knowledge. To formalize this strategy in machine learning, [Bengio *et al.*, 2009] proposed curriculum learning (CL). After that, [Kumar *et al.*, 2010] proposed SPL to achieve the curriculum designing's goal by adding an SPL regularization term into the objective function. Concretely, SPL trains the classification model on "easy" instances first, and then gradually adds "complex" instances to train. The learning difficulty of an instance (whether the instance is "easy" or "complex") is decided by its loss according to the current parameter value. It has been empirically demonstrated that SPL has the ability of avoiding bad local minima and thus has better generalization ability [Kumar *et al.*, 2010; 2011; Jiang *et al.*, 2015; Tang *et al.*, 2012]. Therefore, SPL is typically used to find better solutions for non-convex problems where multiple local minima exist.

Since SR solves a convex optimization problem and can find a global optimum, applying the original SPL model to SR will obtain the same solution as SR. Lately, several variations of SPL were developed to not only select samples to train but also assign a weight to each selected sample. The key step is to apply the soft weighting scheme instead of the hard one in the traditional SPL [Jiang *et al.*, 2014; Zhao *et al.*, 2015; Pi *et al.*, 2016]. In this paper, we utilize such soft weighting technique of SPL to evaluate the importance of each data sample and propose robust softmax regression (RoSR), in which each data sample participates in the training process according to its weight. Typically, noisy data and outliers obtain small weights and their influence will be significantly reduced.

Moreover, in each iteration of SPL, the weights of data instances are determined by those loss values and a global controlling parameter, this might cause that data points of some classes obtain relatively small weights. Hence, these classes have smaller influence than others in the training process and the learned model parameter will be biased. To alleviate this, we propose two novel soft weighting schemes to evaluate weights of data instances locally for each class. This ensures that none of the classes is ignored in the training process and the resulting method is abbreviated to RoSR-L ("L" denotes "Locally").

The main contributions of this work are summarized as follows:

i) To the best of our knowledge, this is the first work to incorporate soft weighting of SPL into softmax regression. The proposed RoSR can significantly reduce the negative influence of noisy data and outliers.

ii) Two soft weighting schemes are developed. The proposed RoSR-L can avoid the situation that some classes are neglected in training and can further improve the performance of RoSR. The methods and experience presented in this paper can be easily used for other classification methods.

iii) Experiments on simulated data show how RoSR and RoSR-L work and extensive experiments on real data are concluded to demonstrate their effectiveness.

2 Related Work

[Kumar *et al.*, 2010] proposed SPL to formulated the curriculum learning's idea as an optimization model by adding a regularization term. SPL learns easy samples first, and then add more samples to training by gradually increasing the controlling parameter. Then, an SPL strategy for the specific-class segmentation task was developed in [Kumar *et al.*, 2011]. [Lee and Grauman, 2011] developed a self-paced discovery framework for visual category discovery tasks. [Tang *et al.*, 2012] formulated a self-paced domain adaptation approach by training target domain knowledge starting with easy samples in the source domain. [SupanČiČ and Ramanan, 2013] designed an SPL method for long term tracking by setting smallest increase in the SVM objective as the loss function. [Jiang *et al.*, 2014] discovered that pseudo relevance feedback is a type of self-paced learning which explains the rationale of this iterative algorithm starting from the easy examples. [Jiang *et al.*, 2015] proposed self-paced curriculum leaning (SPCL), which makes use of prior knowledge in SPL. [Ren *et al.*, 2017] made use of SPL to solve the non-convex problem caused by feature corruption techniques. Traditional SPL methods usually use the hard weighting method, while some work [Jiang *et al.*, 2014; Zhao *et al.*, 2015; Pi *et al.*, 2016] has been done to apply soft weighting schemes. However, all of these existing soft weighting schemes do not consider the difference among the classes and may suffer from the imbalanced class influence problem.

There exists several robust methods of LR have been proposed. Concretely, a distributionally robust method to LR has been developed in [Shafieezadeh-Abadeh *et al.*, 2015]. [Tibshirani and Manning, 2014] presented a novel robust LR approach which adds a term reflecting the possibility of mislabelling in the objective function. [Feng *et al.*, 2014] proposed a robust extension of LR which considers LR with arbitrary outliers in the covariate matrix. To illustrate the influence of robustness, [Feng *et al.*, 2016] proposed robust support vector classifiers (RSVC) for classification problems. Similar with our work, RSVC also tries to reduce the negative influence of noisy data from a weighted viewpoint. The difference is that RSVC aims at solving binary classification problems and the idea of weighting scheme comes from M-estimation in statistic, while our work targets for multi-class classification problems and evaluates the weights of data points via SPL technology.

Although a large amount of work has been presented to address the robustness problem in the presence of noisy data and outliers in binary classification tasks, e.g., LR and support vector machine, none of work has been done to refine SR for multi-class classification from the view of SPL. This work fills the gap by applying the soft weighting scheme of SPL into SR and develops RoSR. Two novel soft weighting scenarios are also developed to further enhance the performance of RoSR.

3 Preliminaries

Let $\mathcal{X} = \{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})\}$ be the data set where $\mathbf{x}^{(i)} \in \mathbb{R}^d$ is the i-th instance, and $y^{(i)} \in \{1, \dots, K\}$ represents its label where $K \geq 2$ is the number of classes. $\mathbf{x}_j^{(i)}$ is the j-th feature of sample $\mathbf{x}^{(i)}$ [1]. The cost between the ground truth label $y^{(i)}$ and the estimated label $g(\mathbf{x}^{(i)}, \boldsymbol{\theta})$ is computed by loss function $L(y^{(i)}, g(\mathbf{x}^{(i)}; \boldsymbol{\theta})) \geq 0$ (abbreviated to l_i). $\boldsymbol{\theta}$ is the model parameter.

3.1 Self-Paced Learning

The goal of self-paced learning (SPL) is to jointly learn $\boldsymbol{\theta}$ and the latent weight variable $\mathbf{v} = [\mathbf{v}_1, \dots, \mathbf{v}_n]^T$ by minimizing:

$$\min_{\boldsymbol{\theta}, \mathbf{v}} J(\boldsymbol{\theta}, \mathbf{v}; \lambda) = \sum_{i=1}^{n} \mathbf{v}_i l_i + f(\lambda, \mathbf{v}) \tag{1}$$

When fix $\mathbf{v}$, Eq. (1) is a traditional classification problem. When fix $\boldsymbol{\theta}$, the solution of $\mathbf{v}$ depends on the definition of the SPL regularization term $f(\lambda, \mathbf{v})$ and the range of values of $\mathbf{v}$. [Kumar *et al.*, 2010] lets $\mathbf{v} \in \{0, 1\}^n$ and defines $f(\lambda, \mathbf{v})$ as

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

[1] The intercept term 1 is added for every instance. Hence, $\mathbf{x}_1^{(i)} = 1, \forall i = 1, \dots, n$.

The optimal $\mathbf{v}^*$ can be calculated by

$$\mathbf{v}_i^* = \begin{cases} 1, & \text{if } l_i < \lambda \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Since $\mathbf{v}_i$ $(i = 1, \dots, n)$ should be equal to either 1 or 0, this method introduced in [Kumar *et al.*, 2010] is considered as hard weighting. $\lambda > 0$ is initially set to a small value, then only those samples with small loss values are selected to train. As λ grows iteratively, more samples will be added until all the samples are chosen.

3.2 Softmax Regression

Let $1\{\cdot\}$ be the indicator function, so that $1\{\text{a true statement}\} = 1$, and $1\{\text{a false statement}\} = 0$. Then the cost function of softmax regression [Bishop, 2006; Bhning, 1992] is

$$J(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} l_i + \mu r(\boldsymbol{\theta}) \tag{4}$$

where the loss function of $\mathbf{x}^{(i)}$ is defined as below:

$$l_i = L(y^{(i)}, g(\mathbf{x}^{(i)}; \boldsymbol{\theta})) = -\sum_{k=1}^{K} 1\{y^{(i)} = k\} \log \frac{e^{\boldsymbol{\theta}_k^T \mathbf{x}^{(i)}}}{\sum_{l=1}^{K} e^{\boldsymbol{\theta}_l^T \mathbf{x}^{(i)}}} \tag{5}$$

The regularization term is

$$r(\boldsymbol{\theta}) = \frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{d} \boldsymbol{\theta}_{kj}^2 \tag{6}$$

and μ is the coefficient. Note that $\boldsymbol{\theta}_k$ denotes the parameter vector of the k-th class and $\boldsymbol{\theta}_{kj}$ is the j-th item of $\boldsymbol{\theta}_k$. As is well-known, SR is a convex model which can find a global optimal solution.

4 Robust Softmax Regression

4.1 Problem Formulation

As mentioned before, the performance of SR is sensitive to the presence of noisy data and outliers. This will be empirically verified later in this paper. To address this, in this work, we introduce Robust Softmax Regression (RoSR) model whose main idea is to make use of soft weighting of SPL. The resulting model is actually a weighted version of SR. The common optimization model of RoSR is defined as follows:

$$\min_{\boldsymbol{\theta}, \mathbf{v}} J(\boldsymbol{\theta}, \mathbf{v}; \lambda) = \sum_{i=1}^{n} \mathbf{v_i} l_i + \mu r(\boldsymbol{\theta}) + f(\lambda, \mathbf{v}) \tag{7}$$

where $\mathbf{v} \in [0,1]^n$ evaluates the weights of data instances, μ is the regularization coefficient, l_i and $r(\boldsymbol{\theta})$ are defined in Eqs. (5) and (6), respectively. If $f(\lambda, \mathbf{v})$ is defined by Eq. (2), then $\mathbf{v}_i$ is either 1 or 0. Then, the optimal solution of Eq. (7) is the same as that of SR. The reason is that once all the data instances are selected, $\mathbf{v}_i = 1$ $(\forall i = 1, \dots, n)$ and $f(\lambda, \mathbf{v})$ is a constant. Thus, the solution of Eq. (7) is the same as minimizing Eq. (4), since both of them can find a global optimum. Thus, in order to make Eq. (7) a weighted version of SR (i.e., let $\mathbf{v}_i \in [0, 1]$), we should use some other definitions of $f(\lambda, \mathbf{v})$.

4.2 Optimization

We use the alternative search strategy to solve Eq. (7). Concretely, we iteratively optimize one of the two parameters $\boldsymbol{\theta}$ and $\mathbf{v}$ while keeping the other one fixed. The main task contains the following two steps.

Step 1: Fix $\boldsymbol{\theta}$, update $\mathbf{v}$. With the fixed $\boldsymbol{\theta}$, l_i and $\mu r(\boldsymbol{\theta})$ in Eq. (7) are constant terms. Then, the weight vector $\mathbf{v}$ can be updated by solving:

$$\mathbf{v}^* = \arg\min \sum_{i=1}^{n} \mathbf{v}_i l_i + f(\lambda, \mathbf{v}) \tag{8}$$

RoSR In [Jiang *et al.*, 2014; Zhao *et al.*, 2015; Pi *et al.*, 2016], $\mathbf{v}_i$ is allowed to take any value in $[0, 1]$. The corresponding methods are considered as soft weighting. Here, we use two types of soft weighting methods (i.e., liner weighting and mixture weighting [Jiang *et al.*, 2014]) in RoSR. The experimental results will show that different soft weighting methods perform comparably with each other. The definition of $f(\lambda, \mathbf{v})$ and the corresponding optimal $\mathbf{v}^*$ of linear weighting [Jiang *et al.*, 2014] are given in Eqs. (9) and (10), respectively.

$$f(\lambda, \mathbf{v}) = \lambda(\frac{1}{2}\|\mathbf{v}\|^2 - \sum_{i=1}^{n} \mathbf{v}_i) \tag{9}$$

$$\mathbf{v}_i^* = \begin{cases} -\frac{l_i}{\lambda} + 1, & \text{if } l_i < \lambda \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

[Jiang *et al.*, 2014] defines $f(\lambda, \mathbf{v})$ of mixture weighting as

$$f(\lambda, \mathbf{v}) = -\sum_{i=1}^{n} \zeta \ln(\mathbf{v}_i + \zeta/\lambda) \tag{11}$$

where an extra SPL parameter $\zeta > 0$ is introduced in addition to λ. Then, the corresponding optimal $\mathbf{v}^*$ is given by [Jiang *et al.*, 2014]:

$$\mathbf{v}_i^* = \begin{cases} 1, & \text{if } l_i \le \zeta\lambda/(\zeta + \lambda) \\ 0, & \text{if } l_i \ge \lambda \\ \zeta/l_i - \zeta/\lambda, & \text{otherwise} \end{cases} \tag{12}$$

For simplicity, we do not tune the parameter ζ and always set $\zeta = 0.5 \times \lambda$ no matter which value of λ is taken. Eqs. (10) and (12) tell us that points with small loss values are assigned with large weights. Noisy data points and outliers, and those points locate in the overlapped regions among the classes, typically have large loss values and will gain small weights. In this way, their influence is significantly reduced.

RoSR-L We can observe from Eqs. (10) and (12) that RoSR chooses instances based on their loss values and a global λ. This may suffer from the imbalanced class influence problem. Concretely, in a certain iteration, the situation probably happens that instances from some classes obtain small loss values and will obtain higher weights. Then, these instances will have more influence in learning the model parameter $\boldsymbol{\theta}$. By contract, most instances from some other classes might have relatively large loss values and will obtain lower weights (even equal to 0). A a result, these classes contribute little in learning $\boldsymbol{\theta}$. If so, the learned $\boldsymbol{\theta}$ is heavily biased and will bring negative influence for the following training process.

To alleviate this, we consider assigning weights and selecting instances locally from each class. To this end, we set controlling parameters λ_k for the k-th class and define two novel SPL regularization terms, shown in Eqs. (13) and (17), respectively. The resulting method is named RoSR-L ("L" means "Locally").

The first regularization term proposed here is

$$f(\lambda,\mathbf{v}) = \sum_{k=1}^{K} \lambda_k \sum_{i\in S_k} \left(\frac{1}{2}\mathbf{v}_i{}^2 - \mathbf{v}_i\right) \tag{13}$$

where S_k denotes the set of indices of data points in the k-th class. If $\mathbf{x}^{(i)}$ belongs to the t-th class, then the optimal $\mathbf{v}_i^*$ is given by

$$\mathbf{v}_i^* = \begin{cases} -\frac{l_i}{\lambda_t} + 1, & \text{if } l_i < \lambda_t \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

Eq. (14) is similar with Eq. (10) and is also considered as linear weighting. λ_k controls the weights of instances and the number of selected instances in the k-th class. The proof of Eq. (14) is given by Theorem 1.

Theorem 1. *If $f(\lambda,\mathbf{v})$ is defined by Eq. (13), the optimal $\mathbf{v}^*$ is given by Eq. (14).*

Proof. Substituting Eq. (13) in Eq. (8), we obtain

$$\mathbf{v}^* = \arg\min \sum_{i=1}^{n} \mathbf{v}_i l_i + \sum_{k=1}^{K} \lambda_k \sum_{i\in S_k} \left(\frac{1}{2}\mathbf{v}_i{}^2 - \mathbf{v}_i\right) \tag{15}$$

By setting the partial derivative of the cost function in Eq. (15) w.r.t. $\mathbf{v}_i$ to 0 and suppose $\mathbf{x}^{(i)}$ belongs to the t-th class, we have

$$l_i + \lambda_t(\mathbf{v}_i - 1) = 0 \tag{16}$$

Then, we can easily obtain the optimal $\mathbf{v}_i^* = -\frac{l_i}{\lambda_t} + 1$. Recall that $\mathbf{v}_i \in [0,1]$ and $l_i \geq 0$, we can obtain that the optimal $\mathbf{v}^*$ can be calculated by Eq. (14). □

The second proposed regularization term is defined as

$$f(\lambda,\mathbf{v}) = -\sum_{k=1}^{K} \sum_{i\in S_k} \zeta_k \ln(\mathbf{v}_i + \zeta_k/\lambda_k) \tag{17}$$

The optimal $\mathbf{v}^*$ is computed by (suppose $\mathbf{x}_i$ is a member of the t-th class)

$$\mathbf{v}_i^* = \begin{cases} 1, & \text{if } l_i \leq \zeta_t\lambda_t/(\zeta_t + \lambda_t) \\ 0, & \text{if } l_i \geq \lambda_t \\ \zeta_t/l_i - \zeta_t/\lambda_t, & \text{otherwise} \end{cases} \tag{18}$$

For simplicity and fair comparison, we always set $\zeta_k = 0.5 \times \lambda_k$ in our experiments. Eq. (18) is similar with Eq. (12) and is considered as mixture weighting. The proof of (18) is shown in Theorem 2.

Theorem 2. *If $f(\lambda,\mathbf{v})$ is defined by Eq. (17), the optimal $\mathbf{v}^*$ is given by Eq. (18).*

Proof. Similarly with Theorem 1, we substitute Eq. (17) in Eq. (8) and set the partial derivative of the corresponding cost function w.r.t. $\mathbf{v}_i$ to 0. Then, we have

$$l_i - \frac{\lambda_t\zeta_t}{\lambda_t\mathbf{v}_i + \zeta_t} = 0 \tag{19}$$

proved that $\mathbf{x}_i$ belongs to the t-th class. Then, it can be easily obtained that the optimal $\mathbf{v}_i^* = \zeta_t/l_i - \zeta_t/\lambda_t$. Note again that $\mathbf{v}_i \in [0,1]$ and $l_i \geq 0$, we can find that the optimal $\mathbf{v}^*$ is given by Eq. (18). □

Step 2: Fix $\mathbf{v}$, update $\boldsymbol{\theta}$. When $\mathbf{v}$ is fixed, $f(\lambda,\mathbf{v})$ in Eq. (7) is a constant and we can update $\boldsymbol{\theta}$ by solving:

$$\boldsymbol{\theta}^* = \arg\min \sum_{i=1}^{n} \mathbf{v}_i l_i + \mu r(\boldsymbol{\theta}) \tag{20}$$

Eq. (20) is actually a weighted version of Eq. (4). We use gradient-descent method [Amari *et al.*, 1996] to solve Eq. (20). Then, $\boldsymbol{\theta}_k$ $(k = 1,\ldots,K)$ is iteratively updated by:

$$\boldsymbol{\theta_k} \leftarrow \boldsymbol{\theta_k} - \alpha \left\{ \sum_{i=1}^{n} \mathbf{v}_i \nabla_{\boldsymbol{\theta}_k} l_i + \mu \nabla_{\boldsymbol{\theta}_k} r(\boldsymbol{\theta}) \right\} \tag{21}$$

where $\nabla_{\boldsymbol{\theta}_k} l_i$ and $\nabla_{\boldsymbol{\theta}_k} r(\boldsymbol{\theta})$ can be easily observed. α is the learning rate. The iteration stops when it converges or reaches the maximum number of iterations, which is set to 250 in our experiments. The L-BFGS method [2] is used for this optimization.

Step 1 and step 2 are iteratively implemented. In each iteration, we increase λ_k to select more samples to train. In the model, we set λ_k $(k = 1,\ldots,K)$ such that half the instances (whose weights are bigger than 0) from the k-th class are selected in the first iteration. Then, in every following iteration, we increase λ_k to add 10% more instances from the k-th class. As a consequence, λ_k $(k = 1,\ldots,K)$ are automatically determined and μ is the only parameter in Eq. (7). Our model stops when all the instances are chosen. Then, the corresponding learned θ and $\mathbf{v}$ are the final model parameter and weight vector, respectively.

5 Experiments

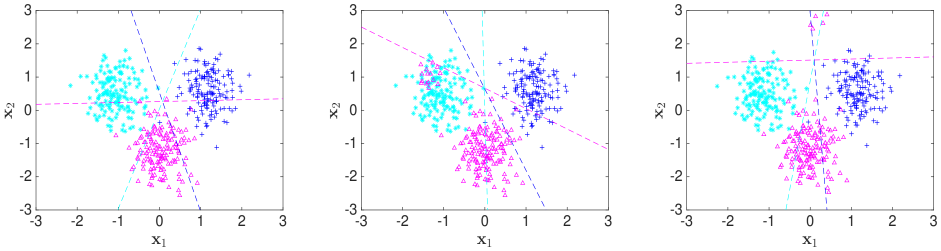
5.1 Experimental Setup

We first conduct experiments on toy examples to show how the proposed methods work. The original toy example (which is shown in Fig. 1 (a)) consists of 3 classes, each providing 150 data points. Thirteen real data sets from different sources are tested in our experiments. The characteristics of these data sets are shown in Table 1. 10digits collects instances for digits 0-9, and each instance is represented by a 28*28 image. CinC (CinC_ECG_torso) is a UCR time series data set [3]. IJCNN1, Seismic, and Vowel are available at https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/. Prostate is a gene expression data set [4]. Balance (Balance

[2] http://www.cs.ubc.ca/∼schmidtm/Software/minFunc.html

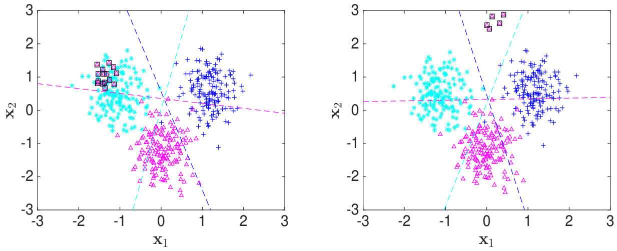
[3] www.cs.ucr.edu/∼eamonn/time_series_data/

[4] http://stat.ethz.ch/∼dettling/bagboost.html



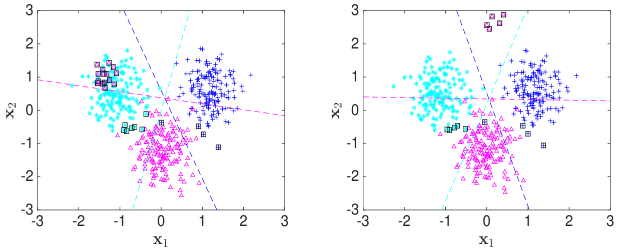
(a) Toy example (b) With noisy data (c) With outliers

Figure 1: Train softmax regression on toy examples.



(a) With noisy data (b) With outliers

Figure 2: Train RoSR-Mix on toy examples. Points whose weights are less than 1 are highlighted with squares.



(a) With noisy data (b) With outliers

Figure 3: Train RoSR-L-Mix on toy examples. Points whose weights are less than 1 are highlighted with squares.

Scale), Biodeg (QSAR biodegradation), Haberman (Haberman's Survival), Letter (Letter Recognition), Musk1 (Musk-Version 1), Skin (Skin Segmentation), and Spambase are all UCI data sets [5]. For each data set, each feature is normalized to have zero mean value and unit variance.

Table 1: Data sets used in the experiments.

Data	#points	#features	#classes
10digits	2000	784	10
Balance	625	4	3
Biodeg	1055	41	2
CinC	1420	1639	4
Haberman	306	3	2
IJCNN1	141691	22	2
Letter	20000	16	26
Musk1	476	166	2
Prostate	102	6033	2
Seismic	98528	50	3
Skin	245057	3	2
Spambase	4601	57	2
Vowel	990	10	11

The following six methods are tested in our experiments:

- LR: Logistic regression with L2 norm.
- SR: Softmax regression [Bhning, 1992].
- RoSR-Lin: RoSR with linear soft weighting.
- RoSR-Mix: RoSR with mixture weighting.
- RoSR-L-Lin: RoSR-L with linear soft weighting.
- RoSR-L-Mix: RoSR-L with mixture weighting.

For RoSR and RoSR-L methods, we first run SR on the whole training data set for 5 iterations to obtain an initialization of the model parameter $\boldsymbol{\theta}$. Actually, each of the six comparing methods has only one regularization coefficient μ needed to be tuned. We tune μ for each method on each data set, where $\mu \in$\{1e-5, 3e-5, 1e-4, 3e-4, 1e-3, 3e-3, 1e-2, 3e-2, 0.1, 0.3, 1\}. For each method, we perform a 10-fold cross validation on each data set ten times and report the average results. We utilize t-test to assess the statistical significance of the results at 5% significance level.

5.2 Results on Toy Examples

Fig. 1(a)-(c) show the three versions of the simulated data set, i.e., the original toy example, the toy example with fifteen noisy data points (those points locate in the area of the cyan "$*$" class but are labeled as the magenta "$\triangle$" class), and the toy example with five outliers (the top five points labeld as the magenta "$\triangle$" class), respectively. In this subsection, the regularization coefficient μ is always set to 1e-3. We train SR on the three data sets and report the corresponding results. Take Fig. 1(a) as example, we obtain three learned model parameter vectors $\boldsymbol{\theta}_i \in \mathbb{R}^3, i = 1, 2, 3$ by SR. Then, we plot $\boldsymbol{\theta}_i^T \mathbf{x} = 0$ (note that $\mathbf{x} = [1, \mathbf{x}_1, \mathbf{x}_2]$ in this case) using a dashed line. The dashed line with a specified color corresponds to the class with the same color. We can observe from Fig. 1(a) that SR works well on the original simulated data set because the three classes are well separated and none of noisy points and outliers exists. However, as shown in Fig. 1(b)-(c), the learned parameters shift a lot when only a small number of noisy data points or outliers exist, indicating the performance of SR is easily affected by noisy data or outliers.

The results of RoSR-Mix and RoSR-L-Mix on toy exam-

[5] http://archive.ics.uci.edu/ml/index.html

Table 2: Results on real data sets (accuracy ± standard derivation, %).

Data	LR	SR	RoSR-Lin	RoSR-Mix	RoSR-L-Lin	RoSR-L-Mix
10digits	85.62±0.30	88.04±0.18	**88.47±0.22**	88.36±0.16	**88.44±0.20**	**88.52±0.26**
Balance	86.37±0.50	89.31±0.36	87.63±0.56	87.71±0.56	**90.94±0.48**	90.72±0.42
Biodeg	87.23±0.17	87.15±0.28	87.78±0.31	**87.95±0.28**	87.86±0.31	**87.90±0.16**
CinC	71.17±0.68	71.39±0.72	72.25±0.51	72.32±0.56	**72.83±0.82**	**73.04±0.70**
Haberman	74.31±0.35	74.31±0.35	74.41±0.49	74.44±0.46	**75.52±0.71**	**75.72±0.46**
IJCNN1	92.28±0.00	92.28±0.00	91.64±0.01	91.51±0.09	**92.68±0.00**	92.50±0.01
Letter	72.39±0.05	77.44±0.06	78.57±0.05	78.22±0.07	**79.15±0.04**	78.98±0.05
Musk1	85.59±1.07	86.03±1.03	**86.72±0.73**	**86.74±0.96**	**86.79±0.65**	**86.87±1.04**
Prostate	92.25±0.56	92.35±0.41	92.35±0.62	92.45±0.66	**93.14±0.46**	**93.14±0.46**
Seismic	70.83±0.01	71.90±0.01	72.36±0.02	72.03±0.01	**72.71±0.01**	72.64±0.01
Skin	91.88±0.00	91.88±0.00	89.95±0.00	90.87±0.00	92.80±0.00	**92.83±0.00**
Spambase	92.16±0.12	92.82±0.11	93.16±0.08	93.12±0.12	**93.22±0.09**	**93.25±0.08**
Vowel	58.17±0.59	67.25±0.22	71.35±0.22	70.98±0.45	**73.18±0.48**	**73.17±0.52**

ples are shown in Figs. 2 and 3, respectively. As expected, Figs. 2 and 3 show that RoSR-Mix and RoSR-L-Mix perform much better than SR in the presence of noisy data and outliers. In Fig. 2, the data points whose weight values assigned by RoSR-Mix are less than 1 are highlighted with squares. It is interesting that all the noisy data and outliers are found. In this manner, the proposed RoSR can also be used for anomaly and mislabelled data detection. Similarly, we highlight those points whose weights are less than 1 with squares in Fig. 3. We can find that more points are highlighted by RoSR-L-Mix than RoSR-Mix. This exactly show the different behaviors between RoSR-L-Mix and RoSR-Mix. RoSR-Mix assigns weights to instances based on their weights globally, while RoSR-L-Mix does this locally for each class, leading to that some points in every class are assigned with lower weights by RoSR-L-Mix. Thus, besides noisy data and outliers, those points in overlapping regions among the classes also play smaller role in the learning process of RoSR-L.

Both RoSR-Mix and RoSR-L-Mix work well on the original toy example. Similar results can also be obtained when applying RoSR and RoSR-L with linear weighting scheme. The corresponding results are not reported due to limited space.

5.3 Results on Real Data Sets

In this subsection, we evaluate the performances of all the comparing classification methods. Table 2 gives the results. In each row, we highlight the best and comparable results in boldface. The first observation from Table 2 is that SR performs comparably with LR on those data sets with only two classes. Besides, SR always achieves higher accuracies than LR. Especially, SR improves upon LR upon by a large margin on 10digits, Balance, Letter, and Vowel. We also find that SR is much faster than LR (the results are not reported due to space limitation) for K-class classification ($K \geq 3$) tasks.

Another observation is that RoSR-Lin and RoSR-Mix always achieve higher accuracies than SR on all the data sets except for Balance, IJCNN1, Prostate, and Skin. This demonstrates the effectiveness of applying SPL with soft weighting schemes in SR. On Prostate, RoSR-Lin performs comparably with SR, while the accuracy achieved by RoSR-Mix is slightly larger than that achieved by SR. Both RoSR-Lin and RoSR-Mix lose to SR on Balance, IJCNN1, and Skin. The main reason is that some classes might be neglected in certain iterations of SPL.

Finally, we can see from Table 2 that RoSR-L-Lin and RoSR-L-Mix perform the best in most cases. The reason for this is two-fold: 1) The RoSR-L methods can significantly reduce the negative influence caused by noisy data and outliers. 2) The SPL strategy that chooses instances to train separately for each class guarantee that all the classes participate in the training process and none of them is neglected in each iteration of SPL.

6 Conclusion and Future Work

In this work we propose robust softmax regression (RoSR) for multi-class classification. RoSR makes use of the soft weighting scheme of self-paced learning to evaluate the importance of each data sample. It can perform well in the presence of noisy data and outliers. Moreover, to alleviate the imbalanced class influence problem, we propose RoSR-L which estimates the weights and selects data samples locally for each class to further enhance the performance of RoSR. Correspondingly, two novel soft weighting schemes are developed. The effectiveness of the proposed methods is demonstrated by experimental results on both synthetic and real data. We are interested in extending the framework proposed in this paper to other machine learning methods in our future work, e.g., kernel methods and neural networks.

Acknowledgments

This paper was in part supported by Grants from the Natural Science Foundation of China (No. 61572111), the National High Technology Research and Development Program of China (863 Program) (No. 2015AA015408), a Project funded by China Postdoctoral Science Foundation (No. 2016M602674), a 985 Project of UESTC (No. A1098531023601041), and two Fundamental Research Funds for the Central Universities of China (Nos. ZYGX2016J078 and ZYGX2016Z003).

References

[Amari *et al.*, 1996] Shun Ichi Amari, Andrzej Cichocki, and Howard Hua Yang. A new learning algorithm for blind signal separation. In *Advances in Neural Information Processing Systems*, pages 757–763, 1996.

[Anand *et al.*, 1993] Rangachari Anand, Kishan Mehrotra, Chilukuri K. Mohan, and Sanjay Ranka. An improved algorithm for neural network classification of imbalanced training sets. *IEEE Transactions on Neural Networks*, 4(6):962–969, 1993.

[Bengio *et al.*, 2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning*, pages 41–48, 2009.

[Bishop, 2006] Christopher M. Bishop. *Pattern Recognition and Machine Learning*, pages 206–210. Springer, 2006.

[Bhning, 1992] Dankmar Bhning. Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics*, 44(1):197–200, 1992.

[Cox, 1958] David R. Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society*, 20(2):215–242, 1958.

[Farid *et al.*, 2014] Dewan Md. Farid, Li Zhang, Chowdhury Mofizur Rahman, M. A. Hossain, and Rebecca Strachan. Hybrid decision tree and naive bayes classifiers for multi-class classification tasks. *Expert Systems with Applications*, 41(4):1937–1946, 2014.

[Feng *et al.*, 2014] Jiashi Feng, Huan Xu, Shie Mannor, and Shuicheng Yan. Robust logistic regression and classification. In *Advances in Neural Information Processing Systems*, pages 253–261, 2014.

[Feng *et al.*, 2016] Yunlong Feng, Yuning Yang, Xiaolin Huang, Siamak Mehrkanoon, and Johan A. K Suykens. Robust support vector machines for classification with nonconvex and smooth losses. *Neural Computation*, 28(6):1217–1247, 2016.

[Freedman, 2009] David A. Freedman. *Statistical models: theory and practice*. Cambridge university press, 2009.

[Hsu and Lin, 2002] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(4):1026–1027, 2002.

[Jiang *et al.*, 2014] Lu Jiang, Deyu Meng, Teruko Mitamura, and Alexander G. Hauptmann. Easy samples first: Self-paced reranking for zero-example multimedia search. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 547–556. ACM, 2014.

[Jiang *et al.*, 2015] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G. Hauptmann. Self-paced curriculum learning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2694–2900, 2015.

[Krawczyk *et al.*, 2014] Bartosz Krawczyk, Micha Woniak, and Gerald Schaefer. Cost-sensitive decision tree ensembles for effective imbalanced classification. *Applied Soft Computing*, 14(1):554–562, 2014.

[Kumar *et al.*, 2010] M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010.

[Kumar *et al.*, 2011] M. Pawan Kumar, Haithem Turki, Dan Preston, and Daphne Koller. Learning specific-class segmentation from diverse data. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1800–1807, 2011.

[Lee and Grauman, 2011] Yong Jae Lee and Kristen Grauman. Learning the easy things first: Self-paced visual category discovery. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1721–1728, 2011.

[Pi *et al.*, 2016] Te Pi, Xi Li, Zhongfei Zhang, Deyu Meng, Fei Wu, Jun Xiao, and Yueting Zhuang. Self-paced boost learning for classification. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 1932–1938, 2016.

[Ren *et al.*, 2017] Yazhou Ren, Peng Zhao, Zenglin Xu, and Dezhong Yao. Balanced self-paced learning with feature corruption. In *Proceedings of the International Joint Conference on Neural Networks*, pages 2064–2071, 2017.

[Shafieezadeh-Abadeh *et al.*, 2015] Soroosh Shafieezadeh-Abadeh, Peyman Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. In *Advances in Neural Information Processing Systems*, pages 1576–1584, 2015.

[SupanČiČ and Ramanan, 2013] James Steven SupanČiČ and Deva Ramanan. Self-paced learning for long-term tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2379–2386, 2013.

[Tang *et al.*, 2012] Kevin Tang, Vignesh Ramanathan, Fei-Fei Li, and Daphne Koller. Shifting weights: Adapting object detectors from image to video. In *Advances in Neural Information Processing Systems*, pages 647–655, 2012.

[Tibshirani and Manning, 2014] Julie Tibshirani and Christopher D. Manning. Robust logistic regression using shift parameters. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 124–129, 2014.

[Walker and Duncan, 1967] Strother H. Walker and David B. Duncan. Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1-2):167–179, 1967.

[Zhao *et al.*, 2015] Qian Zhao, Deyu Meng, Lu Jiang, Qi Xie, Zongben Xu, and Alexander G Hauptmann. Self-paced learning for matrix factorization. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 3196–3202, 2015.