On the Convergence of a Family of Robust Losses for Stochastic Gradient Descent

Bo Han, Ivor W. Tsang(✉), and Ling Chen

Centre for Quantum Computation and Intelligent Systems,
University of Technology Sydney, Sydney, Australia
Bo.Han@student.uts.edu.au.com, {Ivor.Tsang,Ling.Chen}@uts.edu.au.com

Abstract. The convergence of Stochastic Gradient Descent (SGD) using convex loss functions has been widely studied. However, vanilla SGD methods using convex losses cannot perform well with noisy labels, which adversely affect the update of the primal variable in SGD methods. Unfortunately, noisy labels are ubiquitous in real world applications such as crowdsourcing. To handle noisy labels, in this paper, we present a family of robust losses for SGD methods. By employing our robust losses, SGD methods successfully reduce negative effects caused by noisy labels on each update of the primal variable. We not only reveal the convergence rate of SGD methods using robust losses, but also provide the robustness analysis on two representative robust losses. Comprehensive experimental results on six real-world datasets show that SGD methods using robust losses are obviously more robust than other baseline methods in most situations with fast convergence.

1 Introduction

To handle large-scale optimization problems, a popular strategy is to employ Stochastic Gradient Descent (SGD) methods because of two advantages. First, they do not need to compute all gradients over the whole dataset in each iteration, which lowers computational cost per iteration. Secondly, they only process a mini-batch of data points [1] or even one data point [2] in each iteration, which vastly reduces the memory storage. Therefore, many researchers have extensively studied and applied various SGD methods [3,4]. For instance, Large-Scale SGD [5] has been substantially applied to the optimization of deep learning models [6]. Primal Estimated Sub-Gradient Solver (Pegasos) [7] is employed to speed up the Support Vector Machines (SVM) methods, which is suitable for large-scale text classification problems. However, vanilla SGD methods suffer from the label noise problem since the noisy labels adversely affect the update of the primal variable in SGD methods. Unfortunately, the label noise problems are

I.W. Tsang—An Australian Future Fellow and Associate Professor with the Centre for Quantum Computation and Intelligent Systems, at the University of Technology Sydney.

P. Frasconi et al. (Eds.): ECML PKDD 2016, Part I, LNAI 9851, pp. 665–680, 2016.
DOI: 10.1007/978-3-319-46128-1_42

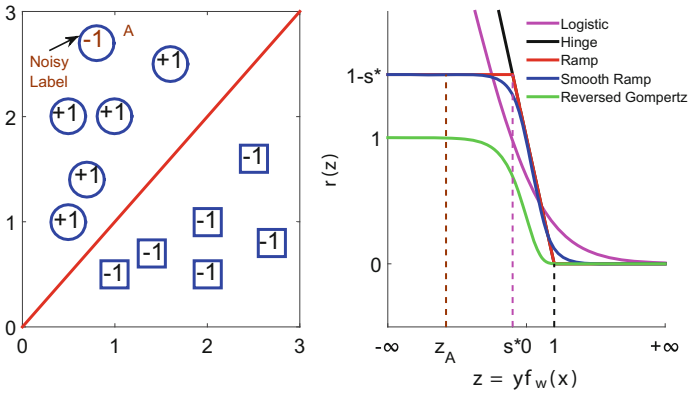

Fig. 1. Left Panel: Squares represent real negative instances. Circles denote real positive instances, however one circle instance "A" is erroneously annotated as negative class, which creates a noisy label. **Right Panel:** Red curve and blue curve respectively denote Ramp Loss and Smooth Ramp Loss parameterized by s^*. Magenta curve, black curve and green curve correspond to Logistic Loss, Hinge Loss and Reversed Gompertz Loss accordingly. It can be observed that the incorrectly labeled instance "A" in the left panel can be regarded as the outlier of negative class, and its loss value $r(z_A)$ is upper bounded by Ramp Loss, Smooth Ramp Loss, and Reversed Gompertz Loss (see "z_A" in the right panel). (Color figure online)

very common in real-world applications. For instance, Amazon Mechanical Turk (MTurk) is a crowdsourcing Internet platform that takes advantage of human intelligence to provide supervision, such as labeling different kinds of bird pictures and annotating keywords according to geoscience records. However, the quality of annotations is not always satisfactory because many workers are not sufficiently trained to label or annotate such specific data [8]. Another situation is where the data labels are automatically inferred from user online behaviors or implicit feedback. For example, the existing recommendation algorithms usually consider a user clicking on an online item (e.g., advertisements on Youtube or eBay) as a positive label indicating user preference, whereas users may click the item for different reasons, such as curiosity or clicking by mistake. Therefore, the labels inferred from online behaviors are often noisy.

The aforementioned issues lead to a challenging question- if the majority of data labels are incorrectly annotated, can we reduce the negative effects on SGD methods caused by these noisy labels? Our high-level idea is to design a robust loss function with a threshold for SGD methods. We illustrate our idea by using a binary classification example. In the left panel of Fig. 1, we notice that the instance $\mathbf{x}_A$ (i.e., data point "A") is incorrectly annotated with the label $y_A = -1$, which is opposite to its predicted label value $(+1)$ according to the hyperplane. Moreover, this instance is far away from the distribution of negative class. Therefore, this instance $\mathbf{x}_A$ with the noisy label y_A can be regarded as the outlier of negative class.

Let the output of the classifier $f_{\mathbf{w}}$ for a given $\mathbf{x}$ be $f_{\mathbf{w}}(\mathbf{x})$. Let z be the product of the real label and the predicted label of an instance $\mathbf{x}$ (i.e., $z = yf_{\mathbf{w}}(\mathbf{x})$). Then, given the outlier $\{\mathbf{x}_A, y_A\}$ in the left panel of Fig. 1, we have $z_A = y_A f_{\mathbf{w}}(\mathbf{x}_A) < 0$. As illustrated in the right panel of Fig. 1, with z on the x-axis, the gradient of Hinge Loss is non-zero on the z_A, which will mislead the update of the primal variable $\mathbf{w}$ in SGD methods. However, if the loss function has a threshold, for example Ramp Loss [9] in Fig. 1 with a threshold $1 - s^*$, the gradient of Ramp Loss on the z_A is zero, which minimizes the negative effects caused by this outlier on the update. Therefore, it is reasonable to employ the loss with a threshold for SGD methods in the label noise problem.

Although the Ramp Loss is robust to outliers, it is computationally hard to optimize due to its nonsmoothness and nonconvexity [10]. Therefore, we consider to relax the Ramp Loss into smooth and locally strongly-convex loss. With random initialization, SGD methods can converge into a qualified local minima with a fast speed. Our main contributions are summarized as follows.

1. We present a family of robust losses, which specifically benefit SGD methods to reduce the negative effects introduced by noisy labels, even under a high percentage of noisy labels.
2. We reveal the convergence rate of SGD methods using the proposed robust losses. Moreover, we provide the robustness analysis on two representative robust losses.
3. Comprehensive experimental results on varying scale datasets with noisy labels show that SGD methods using robust losses are obviously more robust than other baseline methods in most situations with fast convergence.

2 Related Works

First, our work is closely related to SGD methods. For example, Xu proposes the Averaged Stochastic Gradient Descent (ASGD) method [11] to lower the testing error rate of the SGD [5]. However, their work is based on the assumption that the data is clean, which significantly limits their applicability to the label noise problem. Ghahdimi & Lan introduce a randomized stochastic algorithm to solve nonconvex problems [12], and then generalize the accelerated gradient method to improve the convergence rate if the problem is nonconvex [13]. However they do not focus on learning with noisy labels specifically, and do not consider strongly convex regularizer.

Second, our work is also related to bounded nonconvex losses for robust classification. For example, Collobert et al. propose the bounded Ramp Loss for support vector machine (SVM) classification problems. Wang et al. further propose a robust SVM based on a smooth version of Ramp Loss for suppressing the outliers [14]. Their models are commonly inferred by Concave-Convex Procedure (CCCP) [9]. However, both of them do not consider that SGD methods suffer from the label noise problem. In other words, our robust losses are tailor-made for SGD methods to alleviate the effect of noisy labels while their loss is designed only for robust SVM.

Finally, our work is highly related to noisy labels. For instance, Reed & Sukhbaatar focus on training deep neural networks using noisy labels [15]. Natarajan et al. propose a probabilistic model for handling label noise problems [16]. However, all these works are unrelated to SGD methods. Moreover, they cannot be used in real-time or large-scale applications due to their high computational cost. It is also demonstrated that the 0-1 loss function is robust for outliers. However, the 0-1 loss is neither convex nor differentiable, and it is intractable for real learning algorithms in practice. Even though the surrogates of 0-1 loss is convex [17], they are very sensitive to outliers. To the best of our knowledge, the problem of SGD methods for noisy labels has not yet been successfully addressed. This paper therefore studies this problem and provides an answer with theoretical analysis and empirical verification.

3 A Family of Robust Losses for Stochastic Gradient Descent

In this section, we begin with the definition of a family of robust losses for SGD methods. Under this definition, we introduce two representative robust losses: Smooth Ramp Loss and Reversed Gompertz Loss. Then, we reveal the convergence rate of SGD methods using robust losses, and provide the robustness analysis on two representative robust losses.

3.1 Notations and Definitions

Let $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ be the training data, where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the ith instance and $y_i \in \{-1, +1\}$ denotes its binary label. The basic support vector machine model for classification is represented as

$$\min_{\mathbf{w}} G(\mathbf{w}) = \min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} g_i(\mathbf{w}) \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^d$ is the primal variable. Specifically, $g_i(\mathbf{w}) = \rho_\lambda(\mathbf{w}) + r(\mathbf{w}; \{\mathbf{x}_i, y_i\})$ where λ is the regularization parameter, $\rho_\lambda(\mathbf{w})$ is the regularizer and $r(\mathbf{w}; \{\mathbf{x}_i, y_i\})$ is a loss function.

Based on Restricted Strong Convexity (RSC) and Restricted Smoothness (RSM) [18,19], we propose two extended definitions. We use $\|\cdot\|$ to denote the Euclidean norm, and $B_d(\mathbf{w}^*, \gamma)$ to denote the d dimensional Euclidean ball of radius γ centered at local minima $\mathbf{w}^*$. And we assume that function G and g_i are continuously differentiable.

Definition 1 ***(Augmented Restricted Strong Convexity (ARSC)).*** *If there exists a constant $\alpha > 0$ such that for any $\mathbf{w}, \tilde{\mathbf{w}} \in B_d(\mathbf{w}^*, \gamma)$, we have*

$$G(\mathbf{w}) - G(\tilde{\mathbf{w}}) - \langle \nabla G(\tilde{\mathbf{w}}), \mathbf{w} - \tilde{\mathbf{w}} \rangle \geq \frac{\alpha}{2} \|\mathbf{w} - \tilde{\mathbf{w}}\|^2 \tag{2}$$

then G satisfies Augmented Restricted Strong Convexity.

Definition 2 ***(Augmented Restricted Smoothness (ARSM)).*** *If there exists a constant $\beta > 0$ such that for any $i \in \{1, \cdots, n\}$ and $\mathbf{w}, \tilde{\mathbf{w}} \in B_d(\mathbf{w}^*, \gamma)$, we have*

$$g_i(\mathbf{w}) - g_i(\tilde{\mathbf{w}}) - \langle \nabla g_i(\tilde{\mathbf{w}}), \mathbf{w} - \tilde{\mathbf{w}} \rangle \leq \frac{\beta}{2} \|\mathbf{w} - \tilde{\mathbf{w}}\|^2 \tag{3}$$

then g_i satisfies Augmented Restricted Smoothness.

3.2 A Family of Robust Losses

We first present the motivation and definition of a family of robust losses. Take Support Vector Machines (SVM) with convex hinge loss as an example. SGD methods are commonly used to optimize the SVM model for large-scale learning. However, if data points with noisy labels deviate significantly from the hyperplane, these mislabeled data points can be equally viewed as outliers. These outliers will severely mislead the update of the primal variable in SGD methods. Therefore, it is intuitive to design a loss function with a threshold, which truncates the value that exceeds the threshold. Inspired by Ramp Loss [9], we consider whether we can design a family of bounded, locally strongly-convex and smooth losses. If we combine this new loss with strongly-convex regularizer, the objective then satisfies the ARSC (i.e., Definition 1) and ARSM (i.e., Definition 2) simultaneously. Here, we define a family of robust losses $r(z)$ for SGD methods, where z is the variable of loss function in the x-axis of Fig. 1.

Definition 3. *A loss function $r(z)$ is robust for SGD methods if it simultaneously meets the following conditions:*

1. *Upper bound condition - it should be bounded such that $\lim_{z \to -\infty} r'(z) = 0$.*
2. *Locally λ-strongly convex condition - it should be locally λ-strongly convex if there exists a constant $\lambda > 0$ such that $r(z) - \frac{\lambda}{2}\|z\|^2$ is convex when $z \in B_1(z^*, \gamma)$, where $B_1(z^*, \gamma)$ denotes the 1 dimensional Euclidean ball of radius $\gamma > 0$ centered at local minima z^*.*
3. *Smoothly decreasing condition - it should be monotonically decreasing and continuously differentiable.*

Remark 1. We explain three conditions on Definition 3. (1) Since the upper bound can be equally viewed as the threshold, it is natural that the negative effects introduced by outliers are removed by the upper bound. (2) The loss function should be locally λ-strongly convex. If the loss function is locally λ-strongly convex and the regularizer is globally λ-strongly convex (e.g., $\frac{\lambda}{2}\|\mathbf{w}\|^2$), the objective $G(\mathbf{w})$ is locally strongly-convex. Then, objective $G(\mathbf{w})$ satisfies the ARSC. (3) If the loss function is monotonically decreasing, we reasonably assume that the objective is non-increasing around some local minima, which is convenient to prove the convergence rate. If the loss function is differentiable at every point, $g_i(\mathbf{w})$ satisfies the ARSM when $\frac{\lambda}{2}\|\mathbf{w}\|^2$ is used.

Then a family of robust losses for SGD methods can be acquired under these conditions. Here, we propose two representative robust losses that perfectly

satisfy the above three conditions. Both of them are presented in Fig. 1 and employed through the whole paper.

The first one is the Smooth Ramp Loss (4), which is the smooth version of Ramp Loss[1]. If we smooth the Ramp Loss around s^* and around 1, it is much easier to optimize and satisfy the ARSM. Therefore, we employ reversed sigmoid function to represent the Smooth Ramp Loss.

$$r(s^*, z) = \frac{1 - s^*}{1 + e^{\alpha_{s^*}(z + \beta_{s^*})}} \tag{4}$$

where we set the s^* of Ramp Loss, then the parameters α_{s^*} and β_{s^*} of Smooth Ramp Loss are determined by minimizing the difference between Smooth Ramp Loss and Ramp Loss.

The second one is the Reversed Gompertz Loss, which is a special case of the Gompertz function and we reverse the Gompertz function by the y-axis.

$$r(c^*, z) = e^{-e^{c^* \cdot z}} \tag{5}$$

where the curve of this loss is controlled by parameter c^*. The aforementioned losses are integrated into the SVM model and SGD methods are employed to update the primal variable $\mathbf{w}$.

By employing two above robust losses, we finally summarize the robust SGD algorithm - Stochastic Gradient Descent with Robust Losses in Algorithm 1. Specifically, the generalized algorithm consists of two special cases. For Stochastic Gradient Descent with Smooth Ramp Loss, the algorithm employs "Set I and Update I". For Stochastic Gradient Descent with Reversed Gompertz Loss, the algorithm employs "Set II and Update II". In practical implementations, we often choose option A and also provide averaging option B.

3.3 Convergence Analysis

When we apply SGD methods to SVM model with proposed robust losses, it converges into the qualified local minima. According to the detailed explanation about the three conditions in Sect. 3.2, the objective $G(\mathbf{w})$ satisfies the ARSC and $g_i(\mathbf{w})$ satisfies the ARSM. Based on the ARSC and ARSM, we can analyze the convergence rate of SGD methods using robust losses. We use $\mathbb{E}[\cdot]$ to denote the expectation.

Theorem 1. *Consider that $G(\mathbf{w})$ satisfies Augmented Restricted Strong Convexity and $g_i(\mathbf{w})$ satisfies Augmented Restricted Smoothness. Define $\mathbf{w}^*$ as a local minima and β as the parameter of Augmented Restricted Smoothness. Assume that learning rate η is sufficient to let $G(\mathbf{w}^{(t)})$ be a non-increasing update. After T iterations, we have*

$$G(\mathbf{w}^{(T)}) - G(\mathbf{w}^*) \leq \frac{\mathbb{E}\left[\|\mathbf{w}^{(0)} - \mathbf{w}^*\|^2\right]}{(2\eta - 12\eta^2\beta) \cdot T}$$

[1] The common optimization method for Ramp Loss is using Concave-Convex Procedure (CCCP). However, CCCP is time-consuming compared to SGD methods.

Algorithm 1. Stochastic Gradient Descent with Robust Losses (**SGDRL**)

Input: $\lambda \geq 0$, s^*, c^*, the learning rate η, the max number of epochs T_{max}, and the training set $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$

Initialize: $\tilde{\mathbf{w}}^{(0)} = \mathbf{0}$

Set: $\begin{cases} I : f(\alpha_{s^*}, \beta_{s^*}, g) = e^{\alpha_{s^*}(g+\beta_{s^*})} \\ II : f(c^*, g) = c^* g - e^{c^* g} \end{cases}$

for $epoch = 1, 2, \ldots, T_{max}$ **do**

  Preprocess: $\mathbf{w}^{(0)} = \tilde{\mathbf{w}}^{(epoch-1)}$ and randomly shuffle n training instances in $\mathcal{D}$

  for $t = 1, \ldots, n$ **do**

    Sequentially pick: $\{\mathbf{x}_{it}, y_{it}\}$ from $\mathcal{D}$, $it \in \{1, ..., n\}$

    Compute: $g(\mathbf{w}^{(t-1)}) = (\langle \mathbf{w}^{(t-1)}, \mathbf{x}_{it} \rangle + b) y_{it}$

    $\mathbf{w}^{(t)} = \begin{cases} I : w^{(t-1)} - \eta\big[\lambda \mathbf{w}^{(t-1)} - (1 - s^*)\alpha_{s^*} \mathbf{x}_{it} y_{it} \frac{f(\alpha_{s^*}, \beta_{s^*}, g(\mathbf{w}^{(t-1)}))}{(1 + f(\alpha_{s^*}, \beta_{s^*}, g(\mathbf{w}^{(t-1)})))^2}\big] \\ II : \mathbf{w}^{(t-1)} - \eta\big[\lambda \mathbf{w}^{(t-1)} - c^* \mathbf{x}_{it} y_{it} e^{f(c^*, g(\mathbf{w}^{(t-1)}))}\big] \end{cases}$

  end

  option A: $\tilde{\mathbf{w}}^{(epoch)} = \mathbf{w}^{(n)}$ or **option B:** $\tilde{\mathbf{w}}^{(epoch)} = \frac{1}{n} \sum_{t=1}^n \mathbf{w}^{(t)}$

end

Output: $\tilde{\mathbf{w}}^{(T_{max})}$

Proof Sketch for Theorem 1

Proof. Due to space constraints, here we focus on key steps, and the detailed proof is in the arXiv version[2]. According to stochastic gradient descent update rule $\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} - \eta \nabla g_{it}(\mathbf{w}^{(t-1)})$ where random number $it \in \{1, ..., n\}$, and $\mathbb{E}\big[\nabla g_{it}(\mathbf{w}^{(t-1)})\big] = \nabla G(\mathbf{w}^{(t-1)})$ by (1), we construct the following inequality

$$\begin{aligned} &\mathbb{E}\big[\|\mathbf{w}^{(t)} - \mathbf{w}^*\|^2\big] \\ &= \mathbb{E}\big[\|\mathbf{w}^{(t-1)} - \mathbf{w}^*\|^2\big] + \eta^2 \mathbb{E}\big[\|\nabla g_{it}(\mathbf{w}^{(t-1)})\|^2\big] \\ &\quad - 2\eta \langle \nabla G(\mathbf{w}^{(t-1)}), \mathbf{w}^{(t-1)} - \mathbf{w}^* \rangle \\ &\leq \mathbb{E}\big[\|\mathbf{w}^{(t-1)} - \mathbf{w}^*\|^2\big] + \eta^2 \mathbb{E}\big[\|\nabla g_{it}(\mathbf{w}^{(t-1)})\|^2\big] \\ &\quad - 2\eta \big[G(\mathbf{w}^{(t-1)}) - G(\mathbf{w}^*)\big] \end{aligned} \tag{6}$$

where the inequality employs the ARSC. Then we construct an auxiliary function $\varphi_i(\mathbf{w})$

$$\varphi_i(\mathbf{w}) = g_i(\mathbf{w}) - g_i(\mathbf{w}^*) - \langle \nabla g_i(\mathbf{w}^*), \mathbf{w} - \mathbf{w}^* \rangle \tag{7}$$

And it is obvious that

$$\varphi_i(\mathbf{w}^*) = g_i(\mathbf{w}^*) - g_i(\mathbf{w}^*) = 0 \tag{8a}$$

$$\nabla \varphi_i(\mathbf{w}) = \nabla g_i(\mathbf{w}) - \nabla g_i(\mathbf{w}^*) \tag{8b}$$

$$\nabla \varphi_i(\mathbf{w}^*) = \nabla g_i(\mathbf{w}^*) - \nabla g_i(\mathbf{w}^*) = 0 \tag{8c}$$

[2] Please search the arXiv version in https://arxiv.org/abs/1605.01623.

Thus, $\mathbf{w}^*$ is local minima of $\varphi_i(\mathbf{w})$ by (8c) and we construct the following inequality from (7)

$$\begin{aligned} 0 = \varphi_i(\mathbf{w}^*) &\leq \min \varphi_i(\mathbf{w} - \gamma \nabla \varphi_i(\mathbf{w})) \\ &\leq \min \varphi_i(\mathbf{w}) + \frac{\beta\gamma^2}{2}\|\nabla\varphi_i(\mathbf{w}))\|^2 - \gamma\|\nabla\varphi_i(\mathbf{w}))\|^2 \\ &= \varphi_i(\mathbf{w}) - \frac{1}{2\beta}\|\nabla\varphi_i(\mathbf{w})\|^2 \end{aligned} \tag{9}$$

where the last inequality satisfies the ARSM and the function is minimized at the parameter $\gamma = \frac{1}{\beta}$. We construct the following inequality based on (7), (8b) and (9)

$$\begin{aligned} &\|\nabla g_i(\mathbf{w}) - \nabla g_i(\mathbf{w}^*)\|^2 \\ &\quad \leq 2\beta\big[g_i(\mathbf{w}) - g_i(\mathbf{w}^*) - \langle \nabla g_i(\mathbf{w}^*), \mathbf{w} - \mathbf{w}^*\rangle\big] \end{aligned} \tag{10}$$

Therefore, we have

$$\begin{aligned} &\mathbb{E}\big[\|\nabla g_i(\mathbf{w}) - \nabla g_i(\mathbf{w}^*)\|^2\big] \\ &\quad \leq 2\beta\big[G(\mathbf{w}) - G(\mathbf{w}^*) - \langle \nabla G(\mathbf{w}^*), \mathbf{w} - \mathbf{w}^*\rangle\big] \\ &\quad \leq 4\beta\big[G(\mathbf{w}) - G(\mathbf{w}^*)\big] \end{aligned} \tag{11}$$

where the second last inequality satisfies the ARSC. Because $\|A + B + C\|^2 \leq 3\|A\|^2 + 3\|B\|^2 + 3\|C\|^2$ and $\mathbf{w}^*$ is a local minima, we have the following inequality with $\nabla G(\mathbf{w}^*) = 0$ and (11)

$$\begin{aligned} &\mathbb{E}\big[\|\nabla g_{it}(\mathbf{w}^{(t-1)})\|^2\big] \\ &\quad \leq 3\mathbb{E}\big[\|\nabla g_{it}(\mathbf{w}^{(t-1)}) - \nabla g_{it}(\mathbf{w}^*)\|^2\big] \\ &\qquad + 3\mathbb{E}\big[\|\nabla g_{it}(\mathbf{w}^*) - \nabla G(\mathbf{w}^*)\|^2\big] + 3\mathbb{E}\big[\|\nabla G(\mathbf{w}^*)\|^2\big] \\ &\quad \leq 12\beta\big[G(\mathbf{w}^{(t-1)}) - G(\mathbf{w}^*)\big] \end{aligned} \tag{12}$$

Therefore, (6) equals to the following inequality

$$\begin{aligned} &\mathbb{E}\big[\|\mathbf{w}^{(t)} - \mathbf{w}^*\|^2\big] \\ &\quad \leq \mathbb{E}\big[\|\mathbf{w}^{(t-1)} - \mathbf{w}^*\|^2\big] + \eta^2\mathbb{E}\big[\|\nabla g_{it}(\mathbf{w}^{(t-1)})\|^2\big] \\ &\qquad - 2\eta\big[G(\mathbf{w}^{(t-1)}) - G(\mathbf{w}^*)\big] \\ &\quad \leq \mathbb{E}\big[\|\mathbf{w}^{(t-1)} - \mathbf{w}^*\|^2\big] \\ &\qquad + (12\eta^2\beta - 2\eta)\big[G(\mathbf{w}^{(t-1)}) - G(\mathbf{w}^*)\big] \end{aligned} \tag{13}$$

Based on (13), when t varies from $1 \cdots T$, we get T inequalities respectively, and then simultaneously add the left hand side and right hand side of T inequalities to get

$$\begin{aligned} \mathbb{E}\big[\|\mathbf{w}^{(T)} - \mathbf{w}^*\|^2\big] &\leq \mathbb{E}\big[\|\mathbf{w}^{(0)} - \mathbf{w}^*\|^2\big] \\ &+ (12\eta^2\beta - 2\eta)\big[\sum_{t=1}^{T} G(\mathbf{w}^{(t-1)}) - T \cdot G(\mathbf{w}^*)\big] \end{aligned} \tag{14}$$

Under the assumption of a non-increasing update, we have the following inequality

$$\begin{aligned}
&(2\eta - 12\eta^2\beta)\big[T \cdot G(\mathbf{w}^{(T)}) - T \cdot G(\mathbf{w}^*)\big] \\
&\leq (2\eta - 12\eta^2\beta)\big[\sum_{t=1}^{T} G(\mathbf{w}^{(t-1)}) - T \cdot G(\mathbf{w}^*)\big] \\
&\leq \mathbb{E}\big[\|\mathbf{w}^{(0)} - \mathbf{w}^*\|^2\big] - \mathbb{E}\big[\|\mathbf{w}^{(T)} - \mathbf{w}^*\|^2\big]
\end{aligned} \tag{15}$$

We thus obtain

$$\begin{aligned}
G(\mathbf{w}^{(T)}) - G(\mathbf{w}^*) &\leq \frac{\mathbb{E}\big[\|\mathbf{w}^{(0)} - \mathbf{w}^*\|^2\big] - \mathbb{E}\big[\|\mathbf{w}^{(T)} - \mathbf{w}^*\|^2\big]}{(2\eta - 12\eta^2\beta) \cdot T} \\
&\leq \frac{\mathbb{E}\big[\|\mathbf{w}^{(0)} - \mathbf{w}^*\|^2\big]}{(2\eta - 12\eta^2\beta) \cdot T} = \frac{d}{\eta \cdot T} = \epsilon
\end{aligned} \tag{16}$$

where $d = \frac{\mathbb{E}\left[\|\mathbf{w}^{(0)} - \mathbf{w}^*\|^2\right]}{(2-12\eta\beta)}$. Therefore we conclude that when $T = \frac{d}{\eta \cdot \epsilon}$, SGD methods using robust losses have ϵ-solution and the convergence rate is $\mathcal{O}(1/T)$. Therefore, to achieve a ϵ-solution, the complexity of Algorithm 1 is $\mathcal{O}(\frac{n \cdot d}{\eta \cdot \epsilon})$.

4 Robustness Analysis

Theorem 2. *Assume that an instance $\mathbf{x}_i$ is annotated with noisy label y_i, which means $y_i(\mathcal{K}_i^T\alpha + b) < 0$. Its corresponding weighted coefficient ϕ_i for Smooth Ramp Loss with $(s^*, \alpha_{s^*}, \beta_{s^*})$ is*

$$\phi_i = \frac{(1-s^*)\alpha_{s^*}\delta e^{\alpha_{s^*}(y_i\mathcal{K}_i^T\alpha + y_i b)}}{(1-(y_i\mathcal{K}_i^T\alpha + y_i b))(1+\delta e^{\alpha_{s^*}(y_i\mathcal{K}_i^T\alpha + y_i b)})^2}$$

for Reversed Gompertz Loss with c^* is

$$\phi_i = \frac{c^* e^{c^*(y_i\mathcal{K}_i^T\alpha + y_i b)} - e^{c^*(y_i\mathcal{K}_i^T\alpha + y_i b)}}{1-(y_i\mathcal{K}_i^T\alpha + y_i b)}$$

if $|f_w(\mathbf{x}_i)| = |(\mathcal{K}_i^T\alpha + b)|$ increases, which means $\mathbf{x}_i$ with noisy label y_i becomes an outlier, then both ϕ_i will definitely decrease. It indicates that the proposed Robust Losses do reduce the negative effects introduced by noisy labels.

Proof Sketch for Theorem 2

Proof. Firstly, we assume that $\{\mathbf{x}_i, y_i\}_{i=1}^k$ is a random subset of training data $\mathcal{D}$ and $f_{\mathbf{w}}$ is the decision function, according to the representer theorem, $z_i = y_i f_{\mathbf{w}}(\mathbf{x}_i) = y_i(\sum_{j=1}^k \mathcal{K}(\mathbf{x}_j, \mathbf{x}_i)\alpha_j + b) = y_i\mathcal{K}_i^T\alpha + y_i b$, where $\alpha = (\alpha_1, \alpha_2, ..., \alpha_k)'$, $\mathcal{K} = (\mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_k)'$ and $\mathcal{K}_i = (\mathcal{K}(\mathbf{x}_1, \mathbf{x}_i), \mathcal{K}(\mathbf{x}_2, \mathbf{x}_i), ..., \mathcal{K}(\mathbf{x}_k, \mathbf{x}_i))'$. $\lambda > 0$ is a regularizer parameter, $\mathcal{K}$ is a mercer kernel and $\mathcal{H}_{\mathcal{K}}$ is a Reproducing Kernel Hilbert Space (RKHS). For a family of robust losses $r(z)$, we define two functions

$\rho(z)$ and $\varrho(z)$ such that $r(z) = \rho(1-z)$ and $\varrho(z) = \frac{\rho'(z)}{z}$. Therefore, our robust model can be presented as

$$\begin{aligned} f_{\mathbf{w}}^* &= \arg\min_{f_{\mathbf{w}}} \frac{1}{k}\sum_{i=1}^{k} r(z_i) + \frac{\lambda}{2}\|f_{\mathbf{w}}\|^2 \\ &= \arg\min_{f_{\mathbf{w}}} \frac{1}{k}\sum_{i=1}^{k} r(f_{\mathbf{w}}(\mathbf{x}_i)\cdot y_i) + \frac{\lambda}{2} f_{\mathbf{w}}^T f_{\mathbf{w}} \\ &= \arg\min_{\alpha,b} \frac{1}{k}\sum_{i=1}^{k} \rho(1 - y_i\mathcal{K}_i^T\alpha - y_i b) + \frac{\lambda}{2}\alpha^T\mathcal{K}\alpha \end{aligned} \tag{17}$$

The last equation satisfies the second condition of robust losses $r(z) = \rho(1-z)$. Due to $\varrho(z) = \frac{\rho'(z)}{z}$, we define coefficient $\phi_i = \varrho(1 - y_i\mathcal{K}_i^T\alpha - y_i b)$, then

$$\rho'(1 - y_i\mathcal{K}_i^T\alpha - y_i b) = (1 - y_i\mathcal{K}_i^T\alpha - y_i b)\phi_i \tag{18}$$

Because our proposed loss is nonconvex, we assume that $(\hat{\alpha}, \hat{b})$ is one of the critical points for above minimization problem (17). Let's set $Q(\alpha, b) = \frac{1}{k}\sum_{i=1}^{k}\rho(1 - y_i\mathcal{K}_i^T\alpha - y_i b) + \frac{\lambda}{2}\alpha^T\mathcal{K}\alpha$, therefore: $\frac{\partial Q(\hat{\alpha},\hat{b})}{\partial\alpha} = 0$ and $\frac{\partial Q(\hat{\alpha},\hat{b})}{\partial b} = 0$. Then, we have two equations below

$$\frac{1}{k}\sum_{i=1}^{k}(1 - y_i\mathcal{K}_i^T\hat{\alpha} - y_i\hat{b})(y_i\mathcal{K}_i)\phi_i - \lambda\mathcal{K}^T\hat{\alpha} = 0 \tag{19}$$

$$\frac{1}{k}\sum_{i=1}^{k}(1 - y_i\mathcal{K}_i^T\hat{\alpha} - y_i\hat{b})y_i\phi_i = 0 \tag{20}$$

The solution $(\hat{\alpha}, \hat{b})$ of Eqs. (19) and (20) can be achieved by solving the following L2-SVM

$$\min_{\alpha,b} \frac{1}{k}\sum_{i=1}^{k}(y_i - \mathcal{K}_i^T\alpha - b)^2\phi_i + \frac{\lambda}{2}\alpha^T\mathcal{K}\alpha \tag{21}$$

When $k = 1$, we solve it by streaming stochastic gradient descent. If $k > 1$, we solve it by mini-batch stochastic gradient descent. Currently, we consider ϕ_i as an important coefficient that affects the update of stochastic dual variable α, and therefore, we analyze robust statistics briefly from coefficient ϕ_i view.

If an instance $\mathbf{x}_i$ is annotated with noisy label y_i, it means that $y_i f_{\mathbf{w}}(\mathbf{x}_i) < 0$. By the representer theorem, we can easily find $y_i(\mathcal{K}_i^T\alpha + b) < 0$ for this instance. We consider $|(\mathcal{K}_i^T\alpha + b)|$ as the degree where this instance is far away from the hyperplane. So we define $\phi_i = \varrho(1 - y_i\mathcal{K}_i^T\alpha - y_i b)$. To analyze the robustness of $r(z)$, we only take Smooth Ramp Loss as an example here due to space constraints. And the robustness analysis of Reversed Gompertz loss can be found in the arXiv version. We define $\delta = e^{\alpha_{s^*}\beta_{s^*}}$ and according to our inference

$$\begin{aligned}\phi_i &= \varrho(1 - y_i\mathcal{K}_i^T\alpha - y_i b) \\ &= \frac{(1-s^*)\alpha_{s^*}\delta e^{\alpha_{s^*}(y_i\mathcal{K}_i^T\alpha + y_i b)}}{(1-(y_i\mathcal{K}_i^T\alpha + y_i b))(1+\delta e^{\alpha_{s^*}(y_i\mathcal{K}_i^T\alpha + y_i b)})^2}\end{aligned} \tag{22}$$

Remark 2. If $\{\mathbf{x}_i, y_i\}$ is an instance with a noisy label $(y_i(\mathcal{K}_i^T\alpha + b) < 0)$, then the mislabeled instance becomes an outlier when $|f_{\mathbf{w}}(\mathbf{x}_i)| = |(\mathcal{K}_i^T\alpha + b)|$ increases. It means this mislabeled instance is far away from the hyperplane. The coefficient ϕ_i will then decrease because $1 - (y_i\mathcal{K}_i^T\alpha + y_i b)$ will increase while $\frac{e^{\alpha_{s^*}(y_i\mathcal{K}_i^T\alpha + y_i b)}}{(1+\delta e^{\alpha_{s^*}(y_i\mathcal{K}_i^T\alpha + y_i b)})^2}$ will decrease. This indicates that the coefficient ϕ_i will decrease with the increase of $|f_{\mathbf{w}}(\mathbf{x}_i)|$ for outlier instance $\mathbf{x}_i$ and does not play a significant role in the update of the dual variable. Therefore, Smooth Ramp Loss can reduce the negative effects introduced by noisy labels.

5 Experiments

In this section, we mainly perform experiments on noisy datasets to verify the convergence and robustness of SGD methods with two representative robust losses. The datasets range from small to large scale. For convenience, we abbreviate SGD with Smooth Ramp Loss as SGD(SRamp) and SGD with Reversed Gompertz Loss as SGD(RGomp) respectively.

5.1 Experimental Settings

All experimental datasets come from the LIBSVM datasets webpage[3]. The statistics of the datasets are summarized in the Table of the arXiv version. Among them, REAL-SIM, COVTYPE, MNIST38 and IJCNN1 are manually split into the training set and testing set by about 4 : 1. We normalize the data by scaling each feature to [0,1]. To generate the datasets with noisy labels, we follow the settings in [16]. Specifically, we proportionally flip the class label of training data. For example, we randomly flip 20 % of data labels from −1 to 1 or 1 to −1, and assume that the data has 20 % of noisy labels. We then repeat the same process to produce 40 % and 60 % of noisy labels on all datasets.

In the experiments, the baseline methods are classified into two categories. The first category consists of SGD methods with different losses ranging from convex losses to robust nonconvex losses, which can verify the convergence and robustness of SGD methods with two representative losses for noisy labels. For example, we choose SGD with Logistic Loss (SGD(Log)), Hinge Loss (SGD(Hinge)) and Ramp Loss (SGD(Ramp)). We also choose ASGD [11] with Logistic Loss (ASGD(Log)) and PEGASOS [7] as baseline methods. For the second category, we compare proposed methods with LIBLINEAR (We abbreviate L2-regularized L2-loss SVM Primal solution as LIBPrimal and Dual solution as LIBDual) due to its wide popularity in large-scale machine learning. All the

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

Table 1. Testing error rate (in %) with standard deviation on datasets without noisy labels. Methods are indicated by "-"due to running out of memory.

Methods	A7A	IJCNN1	REAL-SIM	COVTYPE	MNIST38	SUSY
LIBPRIMAL	**14.99**	8.25	2.57	24.35	5.71	21.34
LIBDUAL	15.02	8.20	2.67	24.25	6.09	35.32
PEGASOS	17.62 ± 1.56	8.50 ± 0.19	3.32 ± 0.06	26.36 ± 1.99	-	-
SGD(Log)	15.16 ± 0.06	9.08 ± 0.48	2.62 ± 0.03	25.07 ± 0.28	5.73 ± 0.09	20.93 ± 0.01
ASGD(Log)	14.99 ± 0.14	8.04 ± 0.04	2.54 ± 0.01	24.38 ± 0.01	$\mathbf{5.54 \pm 0.01}$	20.83 ± 0.09
SGD(Hinge)	15.45 ± 0.09	8.40 ± 0.22	2.69 ± 0.13	24.62 ± 0.54	5.77 ± 0.16	20.89 ± 0.08
SGD(Ramp)	15.54 ± 0.54	8.50 ± 0.03	4.02 ± 0.02	24.22 ± 0.10	6.04 ± 0.08	21.36 ± 0.05
SGD(SRamp)	15.11 ± 0.06	6.49 ± 0.12	2.55 ± 0.03	23.69 ± 0.04	5.76 ± 0.06	$\mathbf{20.81 \pm 0.03}$
SGD(RGomp)	15.10 ± 0.01	$\mathbf{6.45 \pm 0.02}$	$\mathbf{2.45 \pm 0.03}$	$\mathbf{23.29 \pm 0.03}$	5.56 ± 0.01	20.94 ± 0.01

methods are implemented in C++. Experiments are performed on a computer with a 3.20 GHz Inter CPU and 8 GB main memory running on a Windows 7.

The regularization parameter λ is chosen by 10-fold cross validation for all methods in the range of $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$. For SGD methods with different losses, the number of epochs is normally set to 15 for convergence comparison and the primal variable $\mathbf{w}$ is initialized to $\mathbf{0}$. For LIBLINEAR, we set the bias b to 1 and the stopping tolerance ϵ to 10^{-2} for primal solution and 10^{-1} for dual solution by default. For PEGASOS, the number of epochs for convergence is set to $\frac{10}{\lambda}$ by default and the block size k is set to 1 for training efficiency. For SGD(SRamp), the parameter s^* is chosen by 10-fold cross validation in the range of $[-2, 0]$ according to real-world datasets. Therefore, the parameter $(s^*, \alpha_{s^*}, \beta_{s^*})$ is optimized to $(-0.7, 3, -0.15)$, $(-1, 2, -0.03)$ or $(-2, 1.5, 0.5)$. For SGD(RGomp), the parameter c^* is randomly fixed to 2. All the experiments are repeated ten times and the results are averaged over the 10 trials. Methods are indicated by "-"in Table 1 due to running out of memory. Methods are not reported in Figs. 3 and 4 due to running out of memory or too long training time.[4]

5.2 The Performance of Convergence

First, we verify the convergence of SGD methods with two representative losses for noisy labels. Due to the limit of space, we provide the primal objective value of SGD(SRamp) with the number of epochs on representative small-scale IJCNN1 and large-scale SUSY datasets in the arXiv version. We observe that SGD(SRamp) converges within 15 epochs. This observation is consistent with our convergence analysis in Sect. 3.3. Since SGD(SRamp) and SGD(RGomp) are very similar, the convergence curve of SGD(RGomp) is also similar to that of SGD(SRamp). Thus, we do not report the results of SGD(RGomp).

Then, we further observe the convergence comparison of SGD methods with different losses for noisy labels in Fig. 2 where, with the increase of number

[4] On MNIST38 and SUSY datasets, PEGASOS run out of memory, and the training time of LIBDual is several orders of magnitude more than that of other baselines.

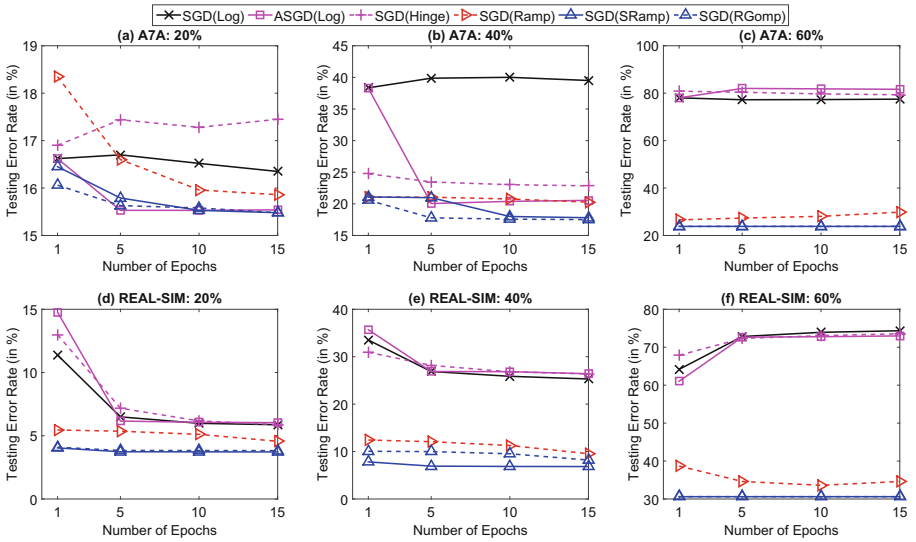

Fig. 2. Testing error rate (in %) with the number of epochs on A7A and REAL-SIM. Datasets have varying percentages (in %) of noisy labels (20 %, 40 % and 60 %). For PEGASOS, the number of epochs for convergence is set to $\frac{10}{\lambda}$ by default. Therefore, we do not report its result

of epochs, the testing error rate of SGD(SRamp) and SGD(RGomp) not only decrease faster than that of other baseline methods but also keep relative stable in the most cases. In other words, our method takes 1–5 epochs to converge while SGD(Hinge) takes more than 15 epochs to converge. Even worse, SGD(Hinge) diverges in presence of 60 % of noisy labels.

5.3 The Performance of Robustness

Finally, we verify the robustness of SGD methods with two representative losses for noisy labels. Figures 3 and 4 respectively report testing error rate and variance with varying percentages of noisy labels. From Figs. 3 and 4, we have the following observations. (a) On all datasets, SGD(SRamp) and SGD(RGomp) obviously outperform the other baseline methods in testing error rate beyond 40 % of noisy labels. Between 0 % to 40 %, SGD(SRamp) and SGD(RGomp) still have comparative advantages. In particular, for a high-dimensional dataset REAL-SIM, the advantage of SGD(SRamp) and SGD(RGomp) is extremely obvious in the whole range of the x-axis. (b) Meanwhile, we notice that the variance of testing error rate for baseline methods (e.g., PEGASOS) gradually increases with the growing percentage of noisy labels, but the variance of testing error rate for SGD(SRamp) and SGD(RGomp) remains at the lowest level in the most cases. Therefore, the robustness of SGD(SRamp) and SGD(RGomp) have been validated by their testing error rate and variance. Although two losses are comparable in the performance of robustness, the parameter of SGD(RGomp) is easier to tune.

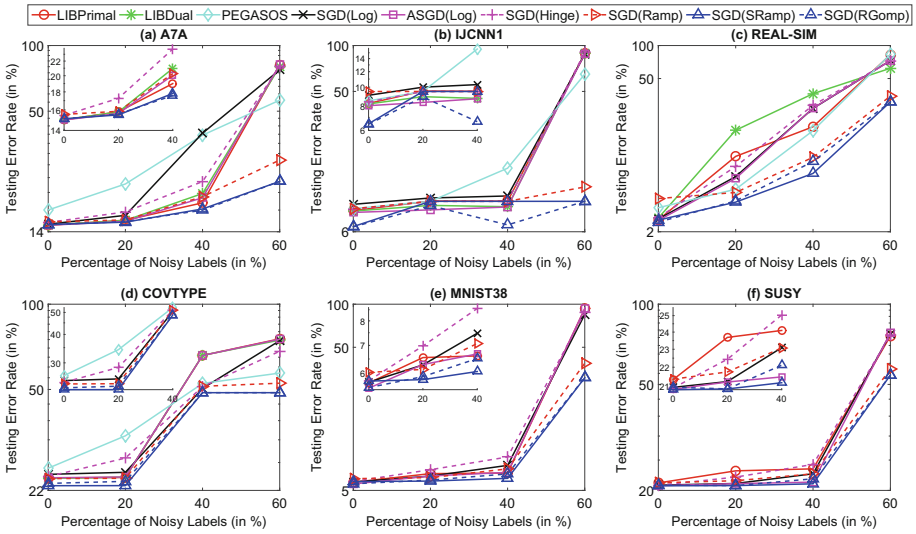

Fig. 3. Testing error rate (in %) on datasets with varying percentages (in %) of noisy labels. We provide the subfigures to compare the testing error rate with 0 % to 40 % of noisy labels on all datasets except for REAL-SIM. The y-axis is in log-scale.

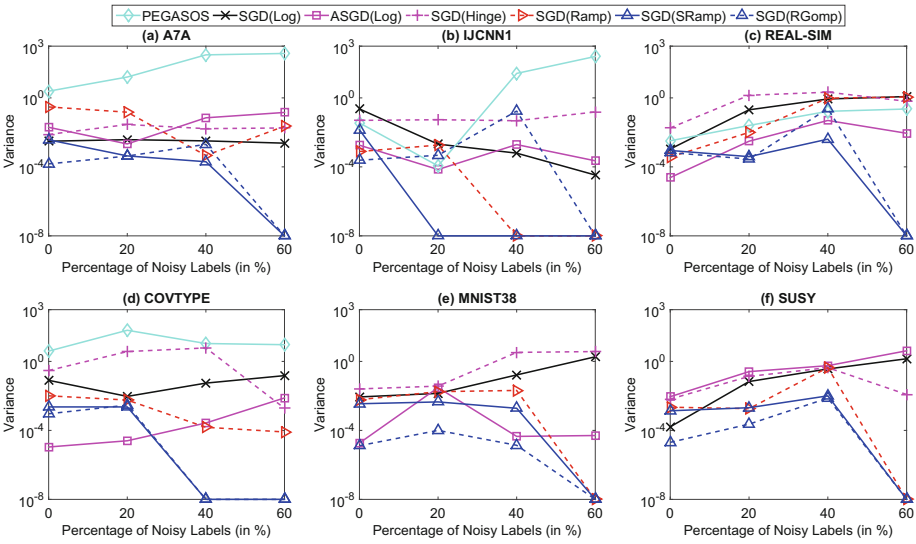

Fig. 4. Variance on datasets with varying percentages (in %) of noisy labels. The y-axis is in log-scale. Note that there is no variance for LIBPrimal and LIBDual because in each update of the primal variable, they compute full gradients instead of stochastic gradients.

In the most cases, the proposed SGD(SRamp) and SGD(RGomp) outperform other baseline methods not only on datasets with varying percentage of noisy labels but also on clean datasets. For example, Table 1 demonstrates that in terms of the testing error rate with the standard deviation, SGD(SRamp) and SGD(RGomp) outperform other baseline methods on IJCNN1, REAL-SIM, COVTYPE and SUSY datasets without noisy labels.

6 Conclusions

This paper studies SGD methods with a family of robust losses for the label noise problem. For convenience, we mainly introduce two representative robust losses including Smooth Ramp Loss and Reversed Gompertz Loss. Our theoretical analysis not only reveals the convergence rate of SGD methods using robust losses, but also proves the robustness of two representative robust losses. Comprehensive experimental results show that, on real-world datasets with varying percentages of noisy labels, SGD methods using our proposed losses are robust enough to reduce negative effects caused by noisy labels with fast convergence. In the future, we will extend our proposed robust losses to improve the performance of SGD methods for regression problems with noisy labels.

Acknowledgments. This work was supported in part by the Australia Research Council (ARC) Discovery Project under Grant No. DP140100545. Dr. Ivor W. Tsang is grateful for the support from the ARC Future Fellowship FT130100746 and ARC Linkage Project under Grant No. LP150100671. Bo Han would like to thank Dr. Chen Gong for many helpful discussions.

References

1. Cotter, A., Shamir, O., Srebro, N., Sridharan, K.: Better mini-batch algorithms via accelerated gradient methods. In: Advances in Neural Information Processing Systems (NIPS), pp. 1647–1655 (2011)
2. Mitliagkas, I., Caramanis, C., Jain, P.: Memory limited, streaming PCA. In: Advances in Neural Information Processing Systems (NIPS), pp. 2886–2894 (2013)
3. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM J. Optim. (SIAM) **22**(2), 341–362 (2012)
4. Agarwal, A., Foster, D.P., Hsu, D., Kakade, S.M., Rakhlin, A.: Stochastic convex optimization with bandit feedback. SIAM J. Optim. (SIAM) **23**(1), 213–240 (2013)
5. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT), pp. 177–187 (2010)
6. Le, Q.V., Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., Ng, A.Y.: On optimization methods for deep learning. In: Proceedings of the 28th International Conference on Machine Learning (ICML), pp. 265–272 (2011)
7. Shalev-shwartz, S., Singer, Y., Srebro, N., Cotter, A.: Pegasos: primal estimated sub-gradient solver for SVM. Math. Program. **127**(1), 3–30 (2011)

8. Yan, Y., Rosales, R., Fung, G., Schmidt, M., Hermosillo, G., Bogoni, L., Moy, L., Dy, J.-G.: Modeling annotator expertise: learning when everybody knows a bit of something. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 932–939 (2010)
9. Collobert, R., Sinz, F., Weston, J., Bottou, L.: Trading convexity for scalability. In: Proceedings of the 23rd International Conference on Machine Learning (ICML), pp. 201–208 (2006)
10. Yu, Y.-L., Yang, M., Xu, L.-L., White, M., Schuurmans, D.: Relaxed clipping: a global training method for robust regression and classification. In: Advances in Neural Information Processing Systems (NIPS), pp. 2532–2540 (2010)
11. Xu, W.: Towards optimal one pass large scale learning with averaged stochastic gradient descent. arXiv preprint arXiv:1107.2490 (2011)
12. Ghadimi, S., Lan, G.-H.: Stochastic first-and zeroth-order methods for nonconvex stochastic programming. SIAM J. Optim. (SIAM) **23**(4), 2341–2368 (2013)
13. Ghadimi, S., Lan, G.-H.: Accelerated gradient methods for nonconvex nonlinear and stochastic programming. Math. Program. **156**, 59–99 (2015)
14. Wang, L., Jia, H.-D., Li, J.: Training robust support vector machine with smooth ramp loss in the primal space. Neurocomputing **71**(13), 3020–3025 (2008)
15. Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., Fergus, R.: Training convolution networks with noisy labels. In: Proceedings of the International Conference on Learning Representations (ICLR) (2015)
16. Natarajan, N., Dhillon, I.S., Ravikumar, P.K., Tewari, A.: Learning with noisy labels. In: Advances in Neural Information Processing Systems (NIPS), pp. 1196–1204 (2013)
17. Bartlett, P.L., Jordan, M.I., McAuliffe, J.D.: Convexity, classification, and risk bounds. J. Am. Stat. Assoc. **101**(473), 138–156 (2006)
18. Agarwal, A., Negahban, S., Wainwright, M.J.: Fast global convergence of gradient methods for high-dimensional statistical recovery. Ann. Stat. **40**(5), 2452–2482 (2012)
19. Loh, P.-L., Wainwright, M.J.: Regularized M-estimators with nonconvexity: statistical and algorithmic theory for local optima. J. Mach. Learn. Res. (JMLR) **16**, 559–616 (2015)