Communications in Information and Systems
Volume 18, Number 1, 37–52, 2018

An investigation for loss functions widely used in machine learning

Feiping Nie, Zhanxuan Hu, and Xuelong Li

Over past few decades, numerous machine learning algorithms have been developed for solving various problems arising in practical applications. And, loss function is one of the most significant factors influencing the performance of algorithm. Nevertheless, most readers may be confused about the reason why these loss functions are effective in corresponding models. The confusion further interfere them to select reasonable loss functions for their algorithms. In this paper, we take a comprehensive investigation for some representative loss functions and analyse the latent properties of them. One of the goals of the investigation is to find the reason why **bilateral** loss functions are more suitable for regression task, while **unilateral** loss functions are more suitable for classification task. In addition, a significant question we discuss is that how to judge the robustness of a loss function. The investigation is useful for readers to develop or improve their future works.

1. Introduction

Numerous machine learning models have been constructed and various taxonomies have been proposed from different perspectives, for instance different learning strategies, different learning types, and so on. Here, we adopt the second taxonomy, i.e., different learning types, and roughly group most existing machine learning algorithms into the following two categories: supervised learning and unsupervised learning.

Classification [9, 13, 27] and regression [3, 24] are two basic tasks in supervised learning. Suppose the input data consists of n samples $\{\boldsymbol{x}_i\}_1^n$ and a target vector $\boldsymbol{y} = (y_1; y_2; \ldots; y_n)$, the goal of supervised learning is to learn a model, i.e., a function $f(\boldsymbol{x})$ which help users predict the value of y for a new sample $\boldsymbol{x}$. The main difference between regression task and classification task is the target value of prediction. Generally, regression model aims to return a **continuous** target value for $\boldsymbol{x}$, while classification model aims to return a **discrete** target value for $\boldsymbol{x}$. In practice, however, the common purpose of them is to learn a function $f(\boldsymbol{x})$ that achieves the minimum loss on all training data. Mathematically, a regression model can be formulated as:

$$\min_{f(\boldsymbol{x})} \sum_{i=1}^{n} l(f(\boldsymbol{x}_i) - y_i) + \mathcal{R}_\lambda(f)\,, \tag{1}$$

where $f(\boldsymbol{x}_i) - y_i$ represents the deviation between $f(\boldsymbol{x}_i)$ and **target value**, $l(r)$ represents a loss function that measures the loss generated by deviation, $\mathcal{R}_\lambda(f)$ is the regularization term for reducing the risk of overfitting. Correspondingly, a binary classification model can be formulated as:

$$\min_{f(\boldsymbol{x})} \sum_{i=1}^{n} l(y_i f(\boldsymbol{x}_i)) + \mathcal{R}_\lambda(f)\,, \tag{2}$$

where y_i is the label of $\boldsymbol{x}_i$, and $y_i f(\boldsymbol{x}_i)$ represents the deviation between $f(\boldsymbol{x}_i)$ and **hyperplane**. For each kind of task, a lot of algorithms have been developed, and loss function is one of the most significant differences between them. For instance, Capped SVR [22] as well as RSVR-GL [26] for support vector regression task, and Capped SVM [19], ramp loss [17, 28], truncated pinball loss [23] and C-Loss [29] for support vector classification task. Facing various loss functions, readers may have the following two questions.

- Question 1: Which types of loss functions can be used to cope with the regression task? And, which types of loss functions can be used to cope with the classification task?
- Question 2: How to judge the robustness of a loss function?

In fact, the second question also occurs in unsupervised learning, such as the problem of low rank approximation. Given a data matrix $\boldsymbol{X}$, a general formulation for low rank approximation is:

$$\min_{\boldsymbol{L}} \|\mathcal{P}(\boldsymbol{L}) - \boldsymbol{X}\|_\ell + \mathcal{R}_\lambda(\boldsymbol{L})\,. \tag{3}$$

where $\mathcal{P}$ is a operator, ℓ denotes the loss function, and $\mathcal{R}_\lambda(\boldsymbol{L})$ denotes the regularization term. The model (3) and its variants have been applied into various fields including Robust principal component analysis (RPCA) [1], matrix completion [2], image denoising [7], and non-rigid structure from moition [5]. In this paper, our investigation is mainly based on the RPCA. The ℓ_2 norm based loss function is generally used in Eq. (3) due to its convexity and smoothness. In practice, however, the ℓ_2-norm based loss function is very sensitive to the outliers, the entries that deviate the optimal target values seriously. Recently, numerous robust loss functions have been developed for solving the setting that input data is corrupted by outliers, such as ℓ_p norm based loss function ($0 < p \leq 1$) [20], and Capped norm based loss function [25]. All of them have improved the robustness of algorithm significantly, but none has provided an explanation or analysis for the improvement. In this paper, we try to fill this blank. We summarize the contribution of this paper as follows:

- By comparing the difference between classification task and regression task, we provide an answer to the Question 1.
- We provide an investigation for loss functions widely used in machine learning, and analyze the robustness of them by comparing some representatives.
- We conduct numerous experiments to verify the conclusions proposed in this paper.

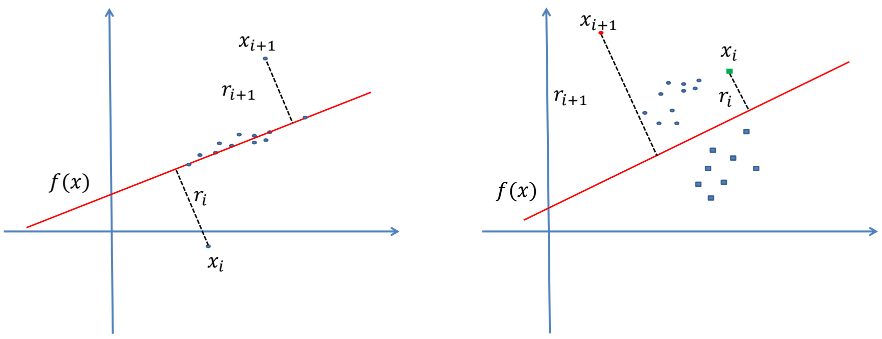
Note that the regularization term is also a significant factor influencing the performances of algorithms mentioned above, but it is not the focus of us in this paper. The rest is organized as follows. In Section 2, we try to answer the question 1. In Section 3, we will discuss the robustness of various loss functions used in machine learning via a low rank approximation model. In Section 4, we implement some experiments to verify the conclusions proposed in this paper. In Sect. 5, we end up this paper by a short discussion.

2. Difference between regression task and classification task

As mentioned above, the main difference between regression task and classification task is the target value of prediction. In this section, we will analyze its impact on the selection of loss function. Note that our investigation is based on two simple models: a linear regression model and a linear classification model. Both of them are very concise, but are convenient for visualization and understanding.

2.1. Regression task

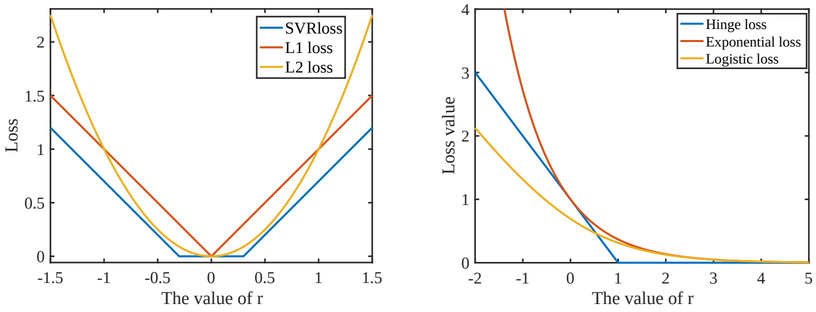
A toy example for linear regression model has been presented in Fig. 1a, where $f(x) = \boldsymbol{w}^T\boldsymbol{x} + b$ is the current regression model learned from training data. The deviation between $f(\boldsymbol{x}_i)$ and **target value** y_i is denoted by $r_i = \boldsymbol{w}^T\boldsymbol{x}_i + b - y_i$, and the corresponding loss is $l(r_i)$. The Fig. 1a shows that



(a) Linear Regression model. (b) Linear Classification model

Figure 1. Two examples with respect to regression task and classification task. Here, $l(r)$ is a loss function that measures the loss generated by deviation. In regression task, the deviation between $f(\boldsymbol{x}_i)$ and **target value** is denoted by $r_i = f(\boldsymbol{x}_i) - y_i$. In classification task, the deviation between $f(\boldsymbol{x}_i)$ and **hyperplane** is denoted by $r_i = y_i f(\boldsymbol{x}_i)$. In Fig. 1a, $r_{i+1} > 0$ and $r_i < 0$, but both $\boldsymbol{x}_{i+1}$ and $\boldsymbol{x}_i$ should be published, and the punishment level should be equal when $|r_i| = |r_{i+1}|$. Hence, for regression task, we expect that $l(r) = l(-r)$. Obviously, the bilateral loss function is a reasonable choice for regression task. In Fig. 1b, the sample $\boldsymbol{x}_i$ with $r_i < 0$ is an incorrect classification and should be punished, while the sample $\boldsymbol{x}_{i+1}$ with $r_{i+1} > 0$ is a correct classification and should not be punished. Hence, for classification task, we expect that $l(r_{i+1}) \ll l(r_i)$ when $r_i < 0$ and $r_{i+1} > 0$. Obviously, the unilateral loss function is a reasonable choice in classification task.

the value of r_{i+1} is larger than 0, while the value of r_i is smaller than 0. In practice, however, both of them should be punished in regression task. Particularly, without considering the imbalanced loss, the punishments to $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$ should be equal when $|r_i| = |r_{i+1}|$, and the punishment to $\boldsymbol{x}_i$ should be larger than to $\boldsymbol{x}_{i+1}$ when $|r_i| > |r_{i+1}|$. As the punishment level for sample depends on its loss value, in regression task we should select a loss function satisfying the conditions that $l(r) = l(-r)$ as well as $l(|r|)$ is monotonically increasing. Obviously, bilateral loss function, such as $l(r) = r^2$ as presented in Fig. 2a is a reasonable choice. Actually, there are some asymmetric loss functions have been utilized to cope with the regression task, such as expectile loss [18]. For this type of loss functions are suitable for some particular settings, we omit them in this paper.



(a) Bilateral loss function (b) Unilateral loss function

Figure 2. One dimensional illustrations for some representative loss functions widely used in regression task and classification task. The bilateral loss functions are generally used in regression task, and the unilateral loss functions are generally used in classification task. Note that $r = f(\boldsymbol{x}) - y$ in regression task, and $r = yf(\boldsymbol{x})$ in classification task.

2.2. Classification task

We now start to observe the characteristics of a linear classification model. In contrast with regression, it aims to learn a model $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ that can return a discrete *label* value for a new sample. More exactly, given a data point $\boldsymbol{x}$ in a two classification task, the target value of y is determined by:

$$y = \begin{cases} 1\ , & \text{if } \ \boldsymbol{w}^T\boldsymbol{x} + b > 0 \\ -1\ , & \text{else.} \end{cases} \tag{4}$$

A toy example for linear classification model has been presented in Fig. 1b, where $f(x) = \boldsymbol{w}^T\boldsymbol{x} + b$ is the current classification model learned from training data. The labels of samples represented by disc are $+1$ and the labels of samples represented by square are -1. The deviation between $f(\boldsymbol{x}_i)$ and **hyperplane** is denoted by $r_i = y_i f(\boldsymbol{x}_i)$ in classification task, and the corresponding loss is $l(r_i)$. The Fig. 1b shows that the value of r_{i+1} is larger than 0, while the value of r_i is smaller than 0. In practice, however, the sample $\boldsymbol{x}_{i+1}$ is a correct classification and should not be punished, but the sample $\boldsymbol{x}_i$ is an incorrect classification and should be punished. As the punishment level for sample depends on its loss value, in classification task

Table 1. Some representative loss functions widely used in regression task and classification task. Note that $r = f(\boldsymbol{x}) - y$ in regression task, and $r = yf(\boldsymbol{x})$ in classification task. Most of these loss functions can also be used to deal with other tasks.

Type	Loss function	Formulation
Bilateral	squares loss function (l_2 loss)	$l(r) = r^2$
	absolute value loss (l_1 loss)	$l(r) = \lvert r\rvert$
	ℓ_p based loss($0 < p < 1$)	$l(r) = \lvert r\rvert^p$
	SVR loss [24]	$l(r) = max(0, \lvert r\rvert - \varepsilon)$
	Huber loss [14]	$HL\,(r) = \begin{cases} \frac{r^2}{2}, & \lvert r\rvert \le k \\ k\,\lvert r\rvert - \frac{k^2}{2}, & \lvert r\rvert > k \end{cases}$
	Capped ℓ_p-norm($0 < p < 2$) [22]	$l(r) = min(\lvert r\rvert^p, a)$
Unilateral	Hinge loss used in SVM [4]	$l(r) = max(0, 1 - r)$
	Squared Hinge loss	$l(r) = (max(0, 1 - r))^2$
	Capped ℓ_p-norm Hinge [21]	$l(r) = min((max(0, 1 - r))^p, \varepsilon)$
	Exponential loss [12]	$l(r) = exp(-r)$
	Logistic loss [11]	$l(r) = log(1 + e^{-r})$
	Unilateral Huber loss [30]	$l(r) = max(0, a - HL(r))$

we should select a loss functions satisfying the condition that $l(r) \gg 0$ when $r \to -\infty$ as well as $l(r) \to 0$ when $r \to +\infty$. Obviously, the unilateral loss function, such as hinge loss $l(r) = max(1 - r, 0)$ as presented in Fig. 2b is a reasonable choice in classification task.

In summary, bilateral loss functions are more suitable for regression task, while unilateral loss functions are more suitable for classification task. We report some representative loss functions widely used in regression and classification tasks in Tab. 1. Note that these loss functions can also be applied into other tasks. Robustness is one of the main differences between them, and it will be discussed in next section.

3. How to judge the robustness of a loss function

In this section, we will discuss the robustness of various loss functions. Note that taking a comprehensive study for loss functions used in all machine learning fields beyond the scope of this paper. Here, we investigate only some representative loss functions and analyze the robustness of them by studying the problem of low rank approximation. The investigation is also suitable for other machine learning tasks including classification and regression.

In addition to the bilateral loss functions reported in Table 1, recently, various loss functions have been developed and can be used to improve the robustness of low rank approximation models [6, 8, 10, 16]. In practice, however, few people analyze the fundamental reason for the improvement. In this section we try to fill this blank.

We start by presenting a variant of the Eq. (3) as follows:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \|\boldsymbol{U}\boldsymbol{V} - \boldsymbol{X}\|_{\ell} \,. \tag{5}$$

where $\boldsymbol{U} \in R^{m\times r}$ and $\boldsymbol{V} \in R^{r\times n}$ are two low rank matrices, and $\boldsymbol{L}$ can be seen as a product of $\boldsymbol{U}$ and $\boldsymbol{V}$, i.e., $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{V}$. Further, it can be rewritten as:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \sum_{i=1}^{m}\sum_{j=1}^{n} l(r_{ij}) \,. \tag{6}$$

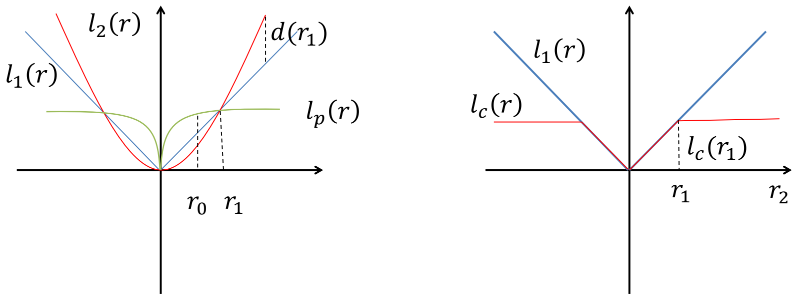
where $r_{ij} = \boldsymbol{u}^i\boldsymbol{v}_j - X_{ij}$ [1] is the deviation between L_{ij} and X_{ij}. $l(x)$ is the loss function that measures the deviation.

Actually, each entry of $\boldsymbol{X}$ can be considered as a target value, and we aim to find a low rank matrix $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{V}$ that achieves the minimum deviation with $\boldsymbol{X}$ on all entries. It is well known that the model generally prefers to punish the entries with larger loss, namely outliers. Note that the outliers are the entries of $\boldsymbol{X}$ that deviate the optimal low rank matrix $\boldsymbol{L}$ significantly in low rank approximation task, and the outliers are the samples that deviate the the optimal model seriously in classification task as well as regression task. Based on the above analysis, we start to discuss the robustness of different loss functions.

- ℓ_2 loss, where $l_2(r) = r^2$;
- ℓ_1 loss, where $l_1(r) = |r|$;
- ℓ_p loss, where $l_p(r) = |r|^p$, where $p = 0.1$;
- Capped ℓ_p loss ($0 < p < 2$), namely l_c loss, where $l_c(r) = min(|r|^p, t)$, and t is the cap we set.

One dimensional illustrations with respect to four loss functions have been presented in Fig. 3, where $p = 0.1$ for l_p loss, and $p = 1$ for l_c loss. Observing the Fig. 3a, we can find the gap between l_1 loss and l_2 loss is monotonously increasing with the increasing of $|r|$, when $|r| > |r_1|$, and the

[1] For a matrix $\boldsymbol{X}$, we denote its i-row and j-th column by $\boldsymbol{x}^i$ and $\boldsymbol{x}_j$, respectively.



(a) (b)

Figure 3. One dimensional illustrations for representative loss functions, where $l_2(r) = r^2$, $l_1(r) = |r|$, $l_p(r) = |r|^p$ $(p = 0.1)$, and $l_c(r) = min(|r|, t)$. Observing (a), we can find that the gap $d(r_1)$ between $l_1(r)$ and $l_2(r)$ increases monotonously with the increasing of r. In addition, $l_p(r) \ll l_1(r) \ll l_2(r)$ when $|r| \to \infty$. Observing (b), we can find that $l_c(r) = l_c(r_1)$ for any $r > r_1$, and the gap between $l_c(r)$ and $l_1(r)$ increases monotonously with the increasing of r. Hence, the robustness of $l_c(r)$ loss to outliers with large magnitude is better than l_1 loss.

loss generated by $l_2(r)$ is far greater than $l_1(r)$ and $l_p(r)$, when r takes a large value. Hence, comparing to the models based on l_1 loss and l_p loss, the models based on l_2 loss will pay more attention to the outliers. For l_p loss, the gap between $l_p(r)$ and $l_p(r_0)$ is very small, even when $r \to \infty$. This property result in the l_p norm based loss function being robust to outliers with large magnitude but sensitive to the noise with small magnitude.

Further, observing the Fig. 3b, we find that the loss generated by l_c is less than l_1 when $|r| > |r_1|$, and $l_c(r) = l_c(r_1)$ for any $|r| > |r_1|$. That is, the loss generated by $l_c(r)$ is not higher than $t = l_c(r_1)$, even when $r \to \infty$, which demonstrates that the impact caused by outliers is limited in l_c loss based model. Hence, the robustness of l_c loss to outliers with high magnitude is better than l_1 loss. Besides, as $l_c(r)$=$l_1(r)$ when $r \leq r_1$, the l_c loss with $p = 1$ is also robust to the outliers with small magnitude.

In summary, the loss function $l(r)$ with an appropriate upper bound when r takes a large value is generally robust to the outliers. In this section, only serval loss functions are discussed, actually, the robustness of other loss functions can also be judged from the viewpoint.

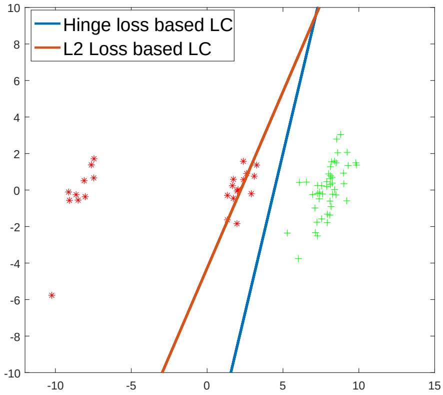


Figure 4. Comparison between two models for Linear Classification (LC) problem. Obviously, the partition result provided by Hinge loss, an unilateral loss function, is better than the partition result provided by ℓ_2 loss, a bilateral loss function. For punishing correct classification and incorrect classification simultaneously, the l_2 loss based model prefers to fit rather than divide the data points.

4. Experiments

In section 4.1, we compare the performances of two models for linear classification task. Note that these two models are based on l_2 loss and Hinge loss, respectively. In Sect. 4.2, we test the robustness of four different loss functions by utilizing them to construct four low rank approximation models. Note that, all experiments are conducted on generated data for the noise level can be controlled arbitrarily.

4.1. Comparison between two models for classification

In this subsection, we use l_2 loss and Hinge loss to construct two linear classification models. In particular, we generate $n = 70$ 2-dimensional points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ as presented in Fig. 4. For each i, the labels of samples with red are $y_i = +1$, and the labels of samples with green are $y_i = -1$. The final results returned by two models are presented in Fig. 4, which shows that the Hinge loss based model can partition data points perfectly, while the l_2 loss based model prefers to fit rather than divide the data points. The reason is that l_2 loss will punish correct classification and incorrect classification simultaneously. Furthermore, we can conclude that using l_2 loss or other bilateral loss functions is inappropriate in classification task.

4.2. Investigate the robustness of different loss functions

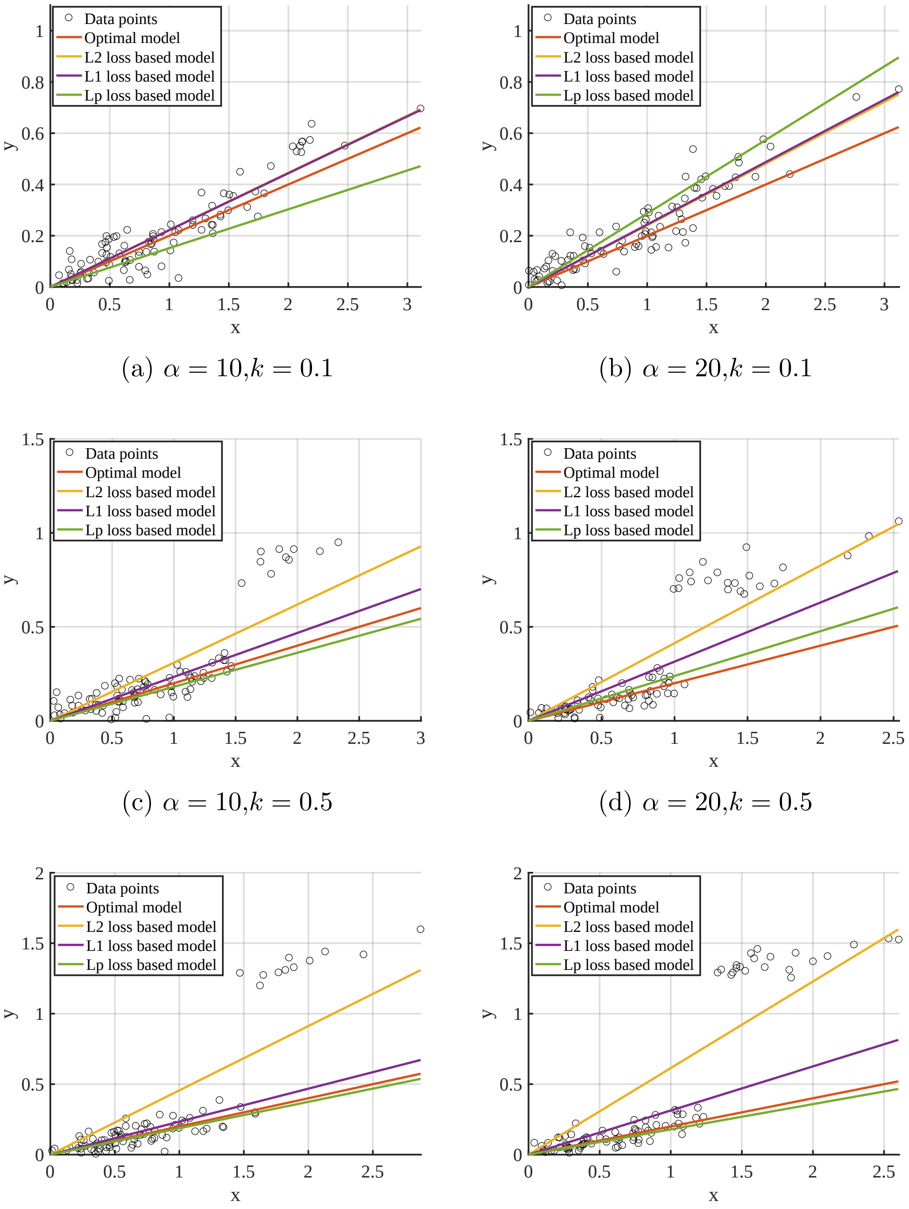
In this subsection, we implement numerous experiments to compare the robustness of four loss functions including l_2 loss, l_1 loss, l_p loss with $p = 0.1$, and l_c loss with $p = 1$. The comparison is based on the problem of low rank approximation. Here, we utilize the formulation (5) to cope with the issue, and utilize four loss functions to generate different models. The optimal solution is provided by conducting SVD when l_2 loss is used. And, for models based on residual loss functions, we utilize the ALM (Augmented Lagrange Multiplier) method [15, 31] to optimize.

4.2.1. Experiments on low dimensional data. In this section we conduct all low rank approximation algorithms, except for the l_c loss based model, on low dimensional data, which is very convenient for visualization. In particular, we generate the input data by the following procedure. First, a matrix $\boldsymbol{L} \in R^{2\times 100}$ is generated, where 100 is the number of samples and 2 is the dimensionality of sample. Suppose $\boldsymbol{l}_i$ is the ith column of matrix $\boldsymbol{L}$, for each i we have:

$$0.2 = \frac{l_{i2}}{l_{i1}}, \tag{7}$$

where l_{ij} denotes the jth element of $\boldsymbol{l}_i$. Thus, we know that $rank(L) = 1$, and its principal direction is $(1, 0.2)$. In addition, note that the entries of the first row of $\boldsymbol{L}$ are sampled from $\mathcal{N}(0, 1)$. Then, we generate two noise matrices $\boldsymbol{E}$ and $\boldsymbol{G}$, where $\boldsymbol{E}$ a sparse noise matrix with $\alpha\%$ entries being k and the residuals being 0, and $\boldsymbol{G}$ is a matrix with entries are sampled from $\mathcal{N}(0, 0.05)$. The noise level are controlled by the values of α and k. Particularly, we vary α in the set $S_\alpha = \{10, 20\}$ and k in the set $S_k = \{0.1, 0.5, 1, 2\}$. The final results are reported in Fig 5, which shows that to reduce the total loss on samples all models we learned will close to the outliers. Observing the Fig. 5a and Fig. 5b, we find that l_p loss is very sensitive to the small noise, and its performance is lower than l_2 loss based loss function. Not only the number but also the magnitude of outliers will degenerate the performances of l_1 loss and l_2 loss based models. While the robustness of l_p loss to noise with large magnitude is prominent.

4.2.2. Experiments on high dimensional data. In this section, we report numerous experimental results on high dimensional data. The low



(a) $\alpha = 10, k = 0.1$

(b) $\alpha = 20, k = 0.1$

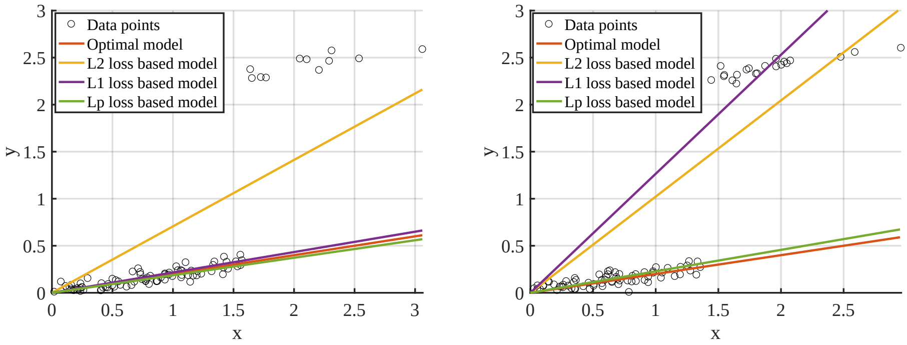
(c) $\alpha = 10, k = 0.5$

(d) $\alpha = 20, k = 0.5$

(e) $\alpha = 10, k = 1$

(f) $\alpha = 20, k = 1$

Figure 5. Experimental results on low dimensional data. Here α controls the number of outliers, k controls the magnitude of outliers.



(a) $\alpha = 10, k = 2$ (b) $\alpha = 20, k = 2$

Figure 5. Experimental results on low dimensional data. Here α controls the number of outliers, k controls the magnitude of outliers.

Table 2. Experimental results on generated data. The l_1 loss is only robust to noise small magnitude, and l_p loss with $p = 0.1$ is only robust to noise large magnitude. The capped l_1 loss, i.e., l_c loss is robust to both of them.

size	Noise level	l_2 loss	l_1 loss	l_p loss	Capped l_1 loss
$m = 100$	$k = 0.01$	$9.72e-04$	$1.09e-08$	$1.35e-03$	$1.03e-08$
	$k = 1$	$9.44e-02$	$9.43e-09$	$5.22e-09$	$9.21e-09$
	$k = 5$	$5.98e-01$	$2.28e-01$	$8.65e-09$	$6.78e-09$
$m = 200$	$k = 0.01$	$9.04e-04$	$1.05e-08$	$1.01e-03$	$1.02e-08$
	$k = 1$	$5.21e-02$	$7.18e-09$	$4.58e-09$	$5.32e-09$
	$k = 5$	$5.35e-01$	$1.01e-01$	$7.37e-09$	$6.21e-09$
$m = 300$	$k = 0.01$	$8.61e-04$	$7.76e-09$	$8.31e-04$	$7.69e-09$
	$k = 1$	$8.52e-02$	$7.83e-09$	$4.25e-09$	$7.58e-09$
	$k = 5$	$4.56e-01$	$8.59e-02$	$6.63e-09$	$1.58e-08$

rank matrix $\boldsymbol{L} \in R^{m \times n}$ with rank r is generated by the following procedure. Firstly, we generate two low rank matrices $\boldsymbol{U} \in R^{m \times r}$ and $\boldsymbol{V} \in R^{r \times n}$ with entries sampled from $\mathcal{N}(0, 1)$, and then generate $\boldsymbol{L}$ by $\boldsymbol{L} = \boldsymbol{UV}$. Subsequently, we generate a sparse matrix $\boldsymbol{E} \in R^{m \times n}$ serve as noise matrix. In particular, 90% entries of $\boldsymbol{E}$ are zero, and the residuals are k. The input noisy matrix is $\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{E}$. In this test, we fix $m = n$, and vary k in the set $S = \{0.01, 1, 5\}$. The precision is measured by RE (Relative Error), which is defined by:

$$RE = \frac{\|\boldsymbol{X}^* - \boldsymbol{L}\|_F^2}{\|\boldsymbol{L}\|_F^2}. \tag{8}$$

where $\boldsymbol{X}^*$ is the solution provided by low rank approximation model. The final results are reported in Tab 2, which shows that l_p (p=0.1) loss is sensitive to the noise with small magnitude, while l_1 as well as l_2 loss are sensitive to the noise with large magnitude. Particularly, the Capped l_1 loss achieves high precision on the all settings. The results are consistent with our analysis mentioned in Sect. 3.

5. Conclusion

This paper provided a comprehensive investigation for loss functions widely used in machine learning. By analysing two simple linear models, we obtain a significant conclusion that the bilateral loss functions are more suitable for regression task, while the unilateral loss functions are more suitable for classification task. Utilizing the proposition that model generally prefers to punish the samples with larger loss, we discussed the robustness of four representative loss functions. Numerical experimental results on toy data demonstrate that the loss functions with an appropriate upper bound value, such as capped norm based loss function when deviation takes a large value, are generally robust to outliers. We mainly discussed three fundamental machine learning models as well as four loss functions in this paper, but most existing complicated algorithms or loss functions can be considered as the variants of them. In addition to loss function, the investigation provided in this paper may be useful for the selection of regularizer, which is also a significant factor influencing the performance of algorithm. The relevant analysis will be presented in our future work. The conclusions summarized in this paper can provide some suggestions to readers for developing or improving their algorithms.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China grant under numbers 61772427 and 61751202.

References

[1] E. J. Candès, X. Li, Y. Ma, and J. Wright, *Robust principal component analysis?*, Journal of the ACM (JACM) **58** (2011), no. 3, 11.

[2] E. J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Foundations of Computational mathematics **9** (2009), no. 6, 717.

[3] R. Collobert and S. Bengio, *SVMTorch: Support vector machines for large-scale regression problems*, Journal of machine learning research **1** (2001 Feb), 143–160.

[4] C. Cortes and V. Vapnik, *Support-vector networks*, Machine learning **20** (1995), no. 3, 273–297.

[5] Y. Dai, H. Li, and M. He, *A simple prior-free method for non-rigid structure-from-motion factorization*, International Journal of Computer Vision **107** (2014), no. 2, 101–122.

[6] C. Gao, N. Wang, Q. Yu, and Z. Zhang, *A Feasible Nonconvex Relaxation Approach to Feature Selection*, in Aaai, 356–361 (2011).

[7] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, *Weighted nuclear norm minimization and its applications to low level vision*, International Journal of Computer Vision **121** (2017), no. 2, 183–208.

[8] N. Guan, T. Liu, Y. Zhang, D. Tao, and L. S. Davis, *Truncated Cauchy Non-negative Matrix Factorization for Robust Subspace Learning*, IEEE Transactions on Pattern Analysis and Machine Intelligence (2017).

[9] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, *Gene selection for cancer classification using support vector machines*, Machine learning **46** (2002), no. 1-3, 389–422.

[10] A. B. Hamza and D. J. Brady, *Reconstruction of reflectance spectra using robust nonnegative matrix factorization*, IEEE Transactions on Signal Processing **54** (2006), no. 9, 3637–3642.

[11] F. E. Harrell, *Ordinal logistic regression*, in: Regression modeling strategies, 331–343, Springer (2001).

[12] T. Hastie, S. Rosset, J. Zhu, and H. Zou, *Multi-class adaboost*, Statistics and its Interface **2** (2009), no. 3, 349–360.

[13] C.-W. Hsu and C.-J. Lin, *A comparison of methods for multiclass support vector machines*, IEEE transactions on Neural Networks **13** (2002), no. 2, 415–425.

[14] P. J. Huber, *Robust Estimation of a Location Parameter*, Annals of Mathematical Statistics **35** (1964), no. 1, 73–101.

[15] Z. Lin, M. Chen, and Y. Ma, *The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices*, `arXiv:1009.5055`, (2010).

[16] C. Lu, J. Tang, M. Lin, L. Lin, S. Yan, and Z. Lin, *Correntropy induced l2 graph for robust subspace clustering*, in Computer Vision (ICCV), 2013 IEEE International Conference on, 1801–1808, IEEE (2013).

[17] H. Masnadi-Shirazi and N. Vasconcelos, *On the design of loss functions for classification: theory, robustness to outliers, and savageboost*, in: Advances in neural information processing systems, 1049–1056 (2009).

[18] W. K. Newey and J. L. Powell, *Asymmetric least squares estimation and testing*, Econometrica: Journal of the Econometric Society (1987), 819–847.

[19] F. Nie, Z. Huo, and H. Huang, *Joint capped norms minimization for robust matrix recovery*, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2557–2563, AAAI Press (2017).

[20] F. Nie, H. Wang, H. Huang, and C. Ding, *Joint Schatten p norm and ℓ_p-norm robust matrix completion for missing value recovery*, Knowledge and Information Systems **42** (2015), no. 3, 525–544.

[21] F. Nie, X. Wang, and H. Huang, *Multiclass capped lp-norm SVM for robust classifications*, in: The 31st AAAI Conference on Artificial Intelligence (AAAI), San Francisco, USA (2017).

[22] A. Safari, *An e–E-insensitive support vector regression machine*, Computational Statistics **29** (2014), no. 6, 1447–1468.

[23] X. Shen, L. Niu, Z. Qi, and Y. Tian, *Support vector machine classifier with truncated pinball loss*, Pattern Recognition **68** (2017), 199–210.

[24] A. J. Smola and B. Schölkopf, *A tutorial on support vector regression*, Statistics and computing **14** (2004), no. 3, 199–222.

[25] Q. Sun, S. Xiang, and J. Ye, *Robust principal component analysis via capped norms*, in Proceedings of the 19th ACM SIGKDD international

conference on Knowledge discovery and data mining, 311–319, ACM (2013).

[26] K. Wang, W. Zhu, and P. Zhong, *Robust support vector regression with generalized loss function and applications*, Neural Processing Letters **41** (2015), no. 1, 89–106.

[27] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, *Hcp: A flexible cnn framework for multi-label image classification*, IEEE transactions on pattern analysis and machine intelligence **38** (2016), no. 9, 1901–1907.

[28] Y. Wu and Y. Liu, *Robust truncated hinge loss support vector machines*, Journal of the American Statistical Association **102** (2007), no. 479, 974–983.

[29] G. Xu, B.-G. Hu, and J. C. Principe, *Robust C-Loss Kernel Classifiers*, IEEE transactions on neural networks and learning systems (2016).

[30] T. Zhang, *Solving large scale linear prediction problems using stochastic gradient descent algorithms*, in Proceedings of the twenty-first international conference on Machine learning **116**, ACM (2004).

[31] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi, *Practical low-rank matrix approximation under robust l 1-norm*, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 1410–1417, IEEE (2012).

School of Computer Science and Center for OPTical IMagery Analysis and Learning
Northwestern Polytechnical University, Xi'an 710072, China
E-mail address: feipingnie@gmail.com

School of Computer Science and Center for OPTical IMagery Analysis and Learning
Northwestern Polytechnical University, Xi'an 710072, China
E-mail address: huzhanxuan@mail.nwpu.edu.cn

Xi'an Institute of Optics and Precision Mechanics
Chinese Academy of Sciences, Xi'an 710119, China
E-mail address: j.lee9383@gmail.com