Robust regression and classification are often thought to require non-convex loss functions that prevent scalable, global training. However, such a view neglects the possibility of reformulated training methods that can yield practically solvable alternatives. A natural way to make a loss function more robust to outliers is to truncate loss values that exceed a maximum threshold. We demonstrate that a relaxation of this form of “loss clipping” can be made globally solvable and applicable to any standard loss while guaranteeing robustness against outliers. We present a generic procedure that can be applied to standard loss functions and demonstrate improved robustness in regression and classification problems.

1 Introduction

Robust statistics is a well established field that analyzes the sensitivity of common estimators to outliers and provides alternative estimators that achieve improved robustness [11, 13, 17, 23]. Outliers are understood to be observations that have been corrupted, incorrectly measured, mis-recorded, drawn under different conditions than those intended, or so atypical as to require separate modeling. The main goal of classical robust statistics is to make estimators invariant, or nearly invariant, to arbitrary changes made to a non-trivial fraction of the sample data—a goal that is equally relevant to machine learning research given that data sets are often collected with limited or no quality control, making outliers ubiquitous. Unfortunately, the state-of-the-art in robust statistics relies on non-convex training criteria that have yet to yield efficient global solution methods [13, 17, 23].

Although many robust regression methods have been proposed in the classical literature, M-estimators continue to be a dominant approach [13, 17]. These correspond to the standard machine learning approach of minimizing a sum of prediction errors under a given loss function (assuming a fixed scaling). M-estimation is reasonably well understood, analytically tractable, and provides a simple framework for trading off between robustness against outliers and data efficiency on inliers [13, 17]. Unfortunately, robustness in this context comes with a cost: when minimizing a convex loss, even a single data point can dominate the result. That is, any (non-constant) convex loss function exhibits necessarily unbounded sensitivity to even a single outlier [17, §5.4.1]. Although unbounded sensitivity can obviously be mitigated by imposing prior bounds on the domain and range of the data [5, 6], such is not always possible in practice. Instead, the classical literature achieves bounded outlier sensitivity by considering redescending loss functions (see [17, §2.2] for a definition), or more restrictively, bounded loss functions, both of which are inherently non-convex. Robust regression has also been extensively investigated in computer vision [2, 26], where a similar conclusion has been reached that bounded loss functions are necessary to counteract the types of outliers created by edge discontinuities, multiple motions, and specularities in image data.

For classification the story is similar. The attempt to avoid outlier sensitivity has led many to propose bounded loss functions [8, 15, 18, 19, 25] to replace the standard convex, unbounded losses deployed in support vector machines and boosting [9] respectively. In fact, [16] has shown that minimizing
any convex margin loss cannot achieve robustness to random misclassification noise. The conclusion reached in the classification literature, as in the regression literature, is therefore that non-convexity is necessary to ensure robustness against outliers—creating an apparent dilemma: one can achieve global training via convexity or outlier robustness via boundedness, but not both.

In this paper we present a counterpoint to these pessimistic conclusions. In particular, we present a general model for bounding any convex loss function, via a process of “loss clipping”, that ensures bounded sensitivity to outliers. Although the resulting optimization problem is not, by itself, convex, we demonstrate an efficient convex relaxation and rounding procedure that guarantees bounded response to data—a guarantee that cannot be established for any convex loss minimization on its own. The approach we propose is generic and can be applied to any standard loss function, be it for regression or classification. Our work is inspired by a number of studies that have investigated robust estimators in computer vision and machine learning [2, 26, 27, 30]. However, these previous attempts were either hampered by local optimization or restricted to special cases; none had guarantees of global training and outlier insensitivity.

Before proceeding it is important to realize that there are many alternative conceptions of “robustness” in the literature that do not correspond to the notion we are investigating. For example, work on “robust optimization” [28, 29] considers minimizing the worst case loss achieved given prespecified bounds on the maximum data deviation that will be considered. Although interesting, these results do not directly bear on the question at hand since we explicitly do not bound the magnitude of the outliers (i.e. the degree of leverage [23, §1.1], nor the size of response deviations). Another notion of robustness is algorithmic stability under leave-one-out perturbation [3]. Although loosely related, algorithmic stability addresses the analysis of given learning procedures rather than describing how a stable algorithm might be generally achieved in the presence of arbitrary outliers. We also do not focus on asymptotic or infinitesimal notions from robust statistics, such as influence functions [11], nor impose boundedness assumptions on the domain and range of the data or the predictor [5, 6].

2 Background

We consider the standard supervised setting where one is given an input matrix $X$ and output targets $y$, with the goal of learning a predictor $h : \mathbb{R}^m \rightarrow \mathbb{R}$. Each row of $X$ gives the feature representation for one training example, denoted $X_i$, with corresponding target $y_i$. We will assume the predictor can be written as a generalized linear model; that is, the predictions are given by $\hat{y}_i = f(X_i, \theta)$ for a fixed transfer function $f$ (possibly identity) and a vector of parameters $\theta$. For training, we will consider the standard $L_2$ regularized loss minimization problem

$$\min_\theta \frac{\gamma}{2} \| \theta \|^2_2 + \sum_{i=1}^n L(y_i, \hat{y}_i) = \min_\theta \frac{\gamma}{2} \| \theta \|^2_2 + \sum_{i=1}^n L(y_i, f(X_i, \theta))$$

(1)

where $L$ denotes the loss function, $\gamma$ is the regularization constant, and $n$ denotes the number of training examples. Normally the loss function $L$ is chosen to be convex in $\theta$ so that the minimization problem can be solved efficiently. Although convexity is important for computational tractability, it has the undesired side-effect of causing unbounded outlier sensitivity, as mentioned. Obviously, the severity of the problem will range from minimal to extreme depending on the nature of the distribution over $(X, y)$. Nevertheless, our goal in this paper will be to eliminate unbounded sensitivity for convex loss functions while retaining a scalable computational approach.\(^1\)

**Standard Convex Loss Functions:** Our general construction applies to arbitrary convex losses, but we will demonstrate our methods on standard loss functions employed in regression and classification. A standard example is Bregman divergences, which are defined by taking a strongly convex differentiable potential $\Phi$ then taking the difference between the potential and its first order Taylor approximation, obtaining a loss $L_\Phi(\tilde{y}||y) = \Phi(\tilde{y}) - \Phi(y) - \phi(y)(\tilde{y} - y)$, where $\phi(y) = \Phi'(y)$ [1, 14]. Several natural loss functions can be defined this way, including least squares $L_\Phi(\tilde{y}||y) = (\tilde{y} - y)^2 / 2$, using the potential $\Phi(y) = y^2 / 2$, and forward KL-divergence $L_\Phi(\tilde{y}||y) = \tilde{y} \ln \frac{\tilde{y}}{y} + (1 - \tilde{y}) \ln \frac{1 - \tilde{y}}{1 - y}$, using the potential $\Phi(y) = y \ln y + (1 - y) \ln(1 - y)$ for $0 \leq y \leq 1$.

\(^1\)All results in this paper extend to reproducing kernel Hilbert spaces via the representer theorem [24], but for clarity of presentation we will use an explicit feature representation $X$ even though it is not a requirement.
A related construction is matching losses [14], which are determined by taking a strictly increasing differentiable transfer function $f$ to be used in prediction via $\hat{y} = f(z)$ where $z = x^\top \theta$. Then, given a transfer $f$, a loss can be defined by \( L_F(\hat{y}||y) = \int^\infty_0 f(\zeta) - f(y) \ dz = F(\hat{y} - F^{-1}(f(z) - f(y) - z) \) such that $F$ satisfies $F'(y) = f(y)$. By definition, matching losses are also Bregman divergences, since $F$ is differentiable and the assumptions on $f$ imply that $F$ is strongly convex. These two loss constructions are related by the equality $L_F(y||\hat{y}) = L_F(\hat{z}||z)$ where $F$ is the Legendre-Fenchel conjugate of $\Phi$ [4, §3.3].

For discrete classification it is also natural to work with a continuous pre-prediction space $\hat{z} = x^\top \theta$, recovering discrete post-predictions $\hat{y} \in \{-1, 1\}$ via a step transfer $\hat{y} = \text{sign}(\hat{z})$. Although a step transfer does not admit the matching loss construction, a surrogate margin loss can be obtained by taking a nonincreasing function $l$ such that $\lim_{m \to \infty} l(m) = 0$, then defining $L_l(\hat{y}, y) = l(\hat{y}y)$. Here $\hat{y}y$ is known as the classification margin. Standard examples include misclassification loss, $L_l(\hat{y}, y) = 1_{(\hat{y}y < 0)}$, and support vector machine (hinge) loss, $L_l(\hat{y}, y) = \max(0, 1 - \hat{y}y)$, binomial deviance loss, $L_l(\hat{y}, y) = \ln(1 + e^{-\hat{y}y})$ [12], and Adaboost loss, $L_l(\hat{y}, y) = e^{-\hat{y}y}$ [9]. If the margin loss is furthermore chosen to be convex, efficient minimization can be attained.

Bounded loss functions: As observed, non-convex loss functions are necessary to bound the ef-
fects of outliers [17]. Black and Rangarajan [2] provide a useful catalog of bounded and redescending loss functions for robust regression, of which a representative example is the Geman and McClure loss $L(y, y) = (\hat{y} - y)^2/(\tau + (\hat{y} - y)^2)$ for $\tau > 0$; see Figure 1. Unfortunately, as Figure 1 makes plain, boundedness implies non-convexity (for any non-constant function). It therefore appears that bounded loss functions achieve robustness at the cost of losing global training guarantees. Our goal is to show that robustness and efficient global training are not mutually exclusive. Despite extensive research on regression and classification, almost no work we are aware of (save perhaps [30] in a limited way) attempts to reconcile robustness to outliers with global training algorithms.

3 Loss Clipping

Adapting the ideas of [2, 27, 30], given any convex loss $\ell(y, x^\top \theta)$ define the clipped loss as

$$\ell_c(y, x^\top \theta) = \min(1, \ell(y, x^\top \theta)).$$

Figure 1 demonstrates loss clipping for some standard loss functions. Given a clipped loss, a robust form of training problem (1) can be written as

$$\min_{\theta} \frac{\gamma}{2} \|\theta\|^2_2 + \sum^n_{i=1} \ell_c(y_i, X_i \theta).$$

Clearly such a training objective bounds the influence of any one training example on the final result. Unfortunately, the formulation (3) is not computationally convenient because the optimization problem it poses is neither convex nor smooth. To make progress on the computational question we exploit a key observation: for any loss function, its corresponding clipped loss can be indirectly expressed by an auxiliary optimization of a smooth objective (if the original loss function itself was smooth). That is, given a loss $\ell(y, x^\top \theta)$ define the corresponding $\rho$-relaxed loss to be

$$\ell_\rho(y, x^\top \theta) = \rho \ell(y, x^\top \theta) + 1 - \rho$$

for $0 \leq \rho \leq 1$; see Figure 1. This construction is an instance of an outlier process as described in [2] and is motivated by a special case hinge-loss construction originally proposed in [30]. The
\( y - \hat{y} \)

\( y \hat{y} \)

\( y \hat{y} \)

\( \rho \)-relaxation provides a convenient characterization of any clipped loss, since it can be shown in general that minimizing a corresponding \( \rho \)-relaxed loss is equivalent to minimizing the clipped loss.

\textbf{Proposition 1} For any loss function \( \ell(y, x^\top \theta) \), we have \( \ell_c(y, x^\top \theta) = \min_{0 \leq \rho \leq 1} \ell(\rho, y, x^\top \theta) \).

(The proof is straightforward, but it is given in the supplement for completeness.) Proposition 1 now allows us to reformulate (3) as a smooth optimization using the fact that the optimization is completely separable between the \( \rho_i \) variables:

\[
\begin{align*}
(3) & \quad = \min_{\theta} \min_{0 \leq \rho \leq 1} \frac{\gamma}{2} \|\theta\|^2 + \sum_{i=1}^{n} \rho_i \ell(y_i, X_i, \theta) + 1 - \rho_i. 
\end{align*}
\]

Unfortunately, the resulting problem is not jointly convex in \( \rho \) and \( \theta \) even though it is convex in each given the other. Such marginal convexity might suggest that an alternating minimization strategy, however the proof of Proposition 1 shows that each minimization over \( \rho \) will result in \( \rho_i = 0 \) for losses greater than 1, or \( \rho_i = 1 \) for losses less than 1. Such discrete assignments immediately causes the search to get trapped in local minima, requiring a more sophisticated approach to be considered.

\section{A Convex Relaxation}

One contribution of this paper is to derive an exact reformulation of (5) that admits a convex relaxation and rounding scheme that retain bounded sensitivity to outliers. We first show how the relaxation can be efficiently solved by a scalable algorithm that eliminates any need for semidefinite programming, then provide a guarantee of bounded outlier sensitivity in Section 5.

\textbf{Reformulation:} To ease the notational burden, let us rewrite (5) in matrix-vector form

\[
(5) = \min_{\rho \leq 1} \min_{\theta} R(\rho, \theta)
\]

where

\[
R(\rho, \theta) = \frac{\gamma}{2} \|\theta\|^2 + \rho \ell(y, X, \theta) + 1 - \rho.
\]

Here \( \mathbf{1} \) denotes the vector of all 1s, and it is understood that \( \ell(y, X, \theta) \) refers to the \( n \times 1 \) vector of individual training losses. Given that \( \ell(\cdot, \cdot, \cdot) \) is convex in its second argument we will be able to exploit Fenchel duality to re-express the min-min form (6) into a min-max form that will serve as the basis for the subsequent relaxation. In particular, consider the definition

\[
\ell^*(y, \alpha) = \sup_{\theta} \alpha \mathbf{1}^\top \theta - \ell(y, \mathbf{1}^\top \theta).
\]

By construction, \( \ell^*(y, \alpha) \) is guaranteed to be convex in \( \alpha \) since it is a pointwise maximum over linear functions [4, §3.2].

\textbf{Lemma 1} For any convex differentiable loss function \( \ell(y, x^\top \theta) \) such that the level sets of \( \ell_\alpha(v) = \alpha x^\top (\theta - v) + \ell(y, x^\top v) \) are bounded, we have

\[
\ell(y, x^\top \theta) = \sup_{\alpha} \alpha x^\top \theta - \ell^*(y, \alpha).
\]

4
(This is a standard result, but a proof is given in the supplement for completeness.) For standard losses \( \ell^*(y, \alpha) \) can be computed explicitly [1, 7]. For example, if \( \ell(y, x^\top \theta) = (y - x^\top \theta)^2/2 \) then \( \ell^*(y, \alpha) = \alpha^2/2 + 3 \alpha y \). Now let \( \Delta(\alpha) \) denote putting \( \alpha \) in the main diagonal of a square matrix and let \( \ell^*(y, \alpha) \) refer to the \( n \times 1 \) vector of dual values over training examples. We can then express the main reformulation as follows.

**Theorem 1** Let \( K = X X^\top \) denote the kernel matrix over input data. Then

\[
\max_{\alpha, \nu} -\frac{1}{n} \sum_{v=1}^{n} \nu_v + \max_{\alpha} \min_{\nu} - (n+1) \nu^\top T(\alpha) \nu \tag{10}
\]

where \( \nu \) is an \( (n+1) \times 1 \) variable, \( \alpha \) is an \( n \times 1 \) variable, and the matrix \( T(\alpha) \) is given by

\[
T(\alpha) = \frac{1}{8n} \begin{bmatrix} 1 \top & \Delta(\alpha) K \Delta(\alpha) [1 \ I] + \frac{1}{4} 
2(1 \top \ell^*(y, \alpha) - n) 
\ell^*(y, \alpha) + 1
\end{bmatrix}. \tag{11}
\]

The proof consists in first dualizing \( \theta \) in (6) via Lemma 1, which establishes the key relationship

\[
\theta = -\frac{1}{7} X^\top \Delta(\rho) \alpha. \tag{12}
\]

The remainder of the proof is merely algebra: given a solution \( \nu \) to (10), the corresponding solution \( \rho \) to (6) can be recovered via \( \rho = \frac{1}{2} (1 + \nu_2/n+1 \sqrt{n+1}) \). See the supplement for full details.

Note that the formulation (10) given in Theorem 1 is exact. No approximation to the problem (6) has been introduced to this point. Unfortunately, as in (6), the formulation (10) is still not directly amenable to an efficient algorithm: the objective is concave in \( \alpha \), conveniently, but it is not convex in \( \nu \). The advantage attained by (10) however is that we can now derive an effective relaxation.

**Optimization Procedure:** Given training data, an outer maximization can be executed jointly over \( \alpha \) and \( \lambda \) to maximize (16). This outer problem is concave in \( \alpha \) and \( \lambda \) hence no local maxima exist. Although the outer problem is not smooth, many effective methods exist for nonsmooth convex optimization [20, 31]. Each outer function evaluation (and subgradient calculation) requires the inner problem (17) to be solved. Fortunately, a simple power method [10] can be used to efficiently compute a maximum eigenvector solution to the inner problem by only performing matrix-vector multiplications on the individual factors of the two low rank matrices making up \( T(\alpha) \), meaning the inner problem can be solved without ever forming a large \( n \times n \) matrix \( T(\alpha) \). That is, if \( X \) is \( n \times m \) each inner iteration requires at most \( O(nm) \) computation.
Solution Recovery: At a solution, the values of (13)–(16) are equal, and all provide a lower bound on the original objective (6). Ideally, given a maximizer \( \alpha^* \) for (14) one would recover a prediction model \( \theta \) via (12). However, (12) requires \( \rho \) to be acquired first, which could be obtained from a \( \nu \) that solves (10). Unfortunately, the relaxation step taken in (13) means that the solution to (14) (recovered from the \( \nu \) that solves (17)) does not necessarily solve (10); the inner solution \( \nu \) in (17) might not be unique. If it is unique, we immediately have the optimal solution to (10) hence an exact solution to the original problem (6). More typically, however, the maximum eigenvector is not unique at \( (\alpha^*, \lambda^*) \), meaning that a gap has been introduced—this occurs if and only if the inner solution \( M^* \) to (14) is not rank 1. In such cases we need to use a rounding procedure to recover an effective rank 1 approximation.

Rounding Method: Given the inner maximizer \( (\alpha^*, \lambda^*) \) of (16) we do not need to explicitly construct the outer minimizer \( M^* \). Instead, it suffices to construct a basis for \( M^* \) by collecting the set of maximum eigenvectors \( \tilde{V} = \{ \tilde{v}_1, ..., \tilde{v}_k \} \) of \( T(\alpha^*) + \Delta(\lambda^*) \) in (17) (note that \( k \) is usually much smaller than \( n + 1 \)). A solution can then be indirectly obtained by solving a small semidefinite program to recover a \( k \times k \) matrix \( C^* \) that satisfies \( C^* \geq 0 \) and \( \delta(V C^* V^\top) = \frac{1}{n+1} \). Note that \( C^* = Q^* \Sigma^* Q^\top \) for some orthonormal \( Q^* \) and diagonal \( \Sigma^* \) where \( \sigma^*_j \geq 0 \) and \( \sum_{j=1}^k \sigma^*_j = 1 \), hence \( M^* = V^* \Sigma^* V^\top \) such that \( V^* = \{ \nu^*_1, ..., \nu^*_k \} = \tilde{V} Q^* \). Given \( V^* \) and \( \Sigma^* \) a rounded solution for \( \hat{\rho} \) can be recovered simply by computing \( \hat{\nu}^* = \sum_{j=1}^k \sigma^*_j \nu^*_j \) and then setting \( \hat{\rho} = \frac{1}{2} \left( 1 + \hat{\nu}^2 n^{-1} \right) \). From the constraints on \( C^* \) it follows that \( \frac{1}{\sqrt{n+1}} \leq \hat{\nu}^*_j \leq \frac{1}{\sqrt{n+1}} \) hence \( 0 \leq \tilde{\rho}_j \leq 1 \forall j \) (details in the supplement). Finally, instead of relying on (12) to recover the model parameters \( \hat{\theta} \) from \( \hat{\rho} \), we explicitly minimize the \( \rho \)-relaxed loss (7) given \( \tilde{\rho} \) to recover \( \hat{\theta} \) via \( \hat{\theta} = \arg\min \hat{R}(\hat{\rho}, \hat{\theta}) \).

Although the rounding step has introduced an approximation, we establish that bounded outlier sensitivity can still be retained, even after the above relaxation and rounding processes, and demonstrate experimentally that the gap from optimality is generally not too large.

5 Bounding Outlier Sensitivity

Thus far we have proposed a robust training objective, provided an efficient convex relaxation that establishes a lower bound, and proposed a simple rounding method for recovering an approximate solution. The question remains as to whether the approximate solution retains bounded sensitivity to outliers (or to leverage points [23, §1.1]). Let \( (\rho^*, \theta^*) \) denote the joint minimizer of (6) and let \( (\hat{\rho}, \hat{\theta}) \) denote the approximate solution obtained from the procedure above.

First, observe that an upper bound on the approximation error can be easily computed by subtracting the lower bound value obtained in (14)–(16) from \( R(\hat{\rho}, \hat{\theta}) \). Our experiments below show that reasonable gaps are obtained in this way. Nevertheless one would still like to guarantee that the gap stays bounded in the presence of arbitrary outliers and leverage points.

Theorem 2 \( \tilde{R}(\hat{\rho}, \alpha^*) \leq 2R(\rho^*, \theta^*) \leq 2n \), where \( \tilde{R}(\hat{\rho}, \alpha^*) \) is the value of (10) at the rounded solution \( \hat{\rho} \). Furthermore, if the uncropped loss \( \ell((y, \hat{y}) \) is \( b\)-Lipschitz in \( \hat{y} \) for \( b < \infty \) and either \( y \) or \( K \) remains bounded, then there exists a \( c < \infty \) such that \( R(\hat{\rho}, \theta) \leq c \).

That is, the \( \rho \)-relaxed loss obtained by the rounded solution stays bounded in this case, even when accounting for the proposed relaxation and rounding procedure and data perturbation. The complete proof takes some space, however the key steps are to show that \( -\frac{1}{n+1} \text{tr}(M^* T(\alpha^*)) \leq \hat{R}(\rho^*, \theta^*) \), and then use this to establish that \( \tilde{R}(\hat{\rho}, \alpha^*) \leq 2R(\rho^*, \theta^*) \) and \( R(\hat{\rho}, \hat{\theta}) \leq c \), respectively (full details in the supplement). Thus, \( (\hat{\rho}, \hat{\theta}) \) will not chase outliers or leverage points arbitrarily in this situation. Note that the proposed method cannot be characterized by minimizing a fixed convex loss. That is, the tightest convex upper bound for any convex loss function is simply given by the function itself, which corresponds to setting \( \rho_i = 1 \) for every training example. By contrast, our approximation method does not choose a constant \( \rho_i = 1 \) for every training example, but instead adaptively chooses \( \rho_i \) values, closer to 1 for inliers and closer to 0 for outliers. The resulting upper bound on the clipped loss (hence on the misclassification error in the margin loss case) is much tighter than that achieved by simply minimizing a convex loss. This outcome is demonstrated clearly in our experiments.
In this section, we experimentally evaluate the preceding technical developments on synthetic and real data for both regression and classification.

**Regression:** We first illustrate the behavior of the various regression techniques by a simple demonstration. In Figure 2 (a) and (b), we generate a cluster of linearly related data $y = x$ in a small interval about the origin, then add outliers. In Figure 2 (c) the target linear model is mixed with another more dispersed model. We compare the behaviours of standard regression losses: least-squares (L2), L1 (L1), the Huber minimax loss (HuberM) [13, 17], and the robust Geman and McClure loss (GM) [2]. To these we compare the proposed relaxed method (ClipRelax), along with an alternating minimizer of the clipped loss (ClipAlt). (In this problem the value of $\gamma$ has little effect, and is simply set to 0.1.) Figure 2 demonstrates that the three convex losses, L2, L1 and HuberM, are dominated by outliers. By contrast, ClipRelax successfully found the correct linear model in each case. Note that the robust GM loss finds two different minima, corresponding to that of L2 and ClipRelax respectively, hence it was not depicted in the plot. ClipAlt also gets trapped in local minima as expected: it finds the correct model in Figure 2 (a) but incorrect models in Figure 2 (b) and (c).

In our second synthetic regression experiment we consider larger problems. Here a target weight vector $\theta$ is drawn from $N(0, I)$, with inputs $X_i$, sampled uniformly from $[0, 1]^m$, $m = 5$. The outputs $y_i$ are computed as $y_i = X_i \theta + \epsilon_i$, $\epsilon_i \sim N(0, \frac{1}{3})$. We then seed the data set with outliers by randomly re-sampling each $y_i$ (and $X_i$) from $N(0, 10^5)$ and $N(0, 10^2)$ respectively, governed by an outlier probability $p$. Here 200 of the 700 examples are randomly chosen as the training set and the rest used for testing. We compare the same six methods: L2, L1, HuberM, GM, ClipAlt and ClipRelax. The regularization parameter $\gamma$ was set on a separate validation set. These experiments are repeated 20 times and average (Huber loss) test errors on clean data are reported (with standard deviations) in Table 1. Clearly, the outliers significantly affect the performance of least squares. In this case the proposed relaxation performs comparably to the the non-convex GM loss. Interestingly, this experiment shows that the relative gap between the $p$-robust loss obtained by the proposed method and the lower bound on the optimal $p$-robust loss (16) remains remarkably small, indicating our robust relaxation (almost) optimally minimizes the original non-convex clipped loss.

### Table 1: Synthetic experiment with $n = 200$, $m = 5$, and $t = 500$. Test error rates (RMSE) on clean data (average ± standard deviations) at different outlier probabilities $p$, 20 repeats. The bottom row shows the relative gap obtained between the $p$-relaxed loss of the rounded solution and the computed lower bound (16).

<table>
<thead>
<tr>
<th>Loss</th>
<th>out. prob.</th>
<th>outlier probability</th>
<th>$p = 0.0$</th>
<th>$p = 0.2$</th>
<th>$p = 0.4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2</td>
<td>2.53 ± 0.0015</td>
<td>25.11 ± 13.78</td>
<td>19.04 ± 15.62</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td>2.53 ± 0.0015</td>
<td>26.52 ± 16.09</td>
<td>27.14 ± 22.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HuberM</td>
<td>2.52 ± 0.0015</td>
<td>12.02 ± 5.33</td>
<td>12.30 ± 5.87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GM (local)</td>
<td>2.53 ± 0.0015</td>
<td>2.60 ± 0.10</td>
<td>2.62 ± 0.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClipAlt (local)</td>
<td>2.53 ± 0.0019</td>
<td>2.75 ± 0.27</td>
<td>2.81 ± 0.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClipRelax</td>
<td>2.53 ± 0.0016</td>
<td>2.68 ± 0.12</td>
<td>2.53 ± 0.87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OptimGap</td>
<td>1.65% ± 0.31%</td>
<td>0.10% ± 0.22%</td>
<td>0.70% ± 1.31%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: Comparison on three demonstration data sets.

In this section, we experimentally evaluate the preceding technical developments on synthetic and real data for both regression and classification.

6 Experimental Results
Finally, we investigated the behavior of the regression methods on a few real data sets. We chose three data sets: astronomy data containing outliers from [23], and two UCI data sets, seeding the the UCI data sets with outliers. Test results are reported on clean data to avoid skewing the reported results. For UCI data, outliers were added by resampling $X_i$ and $y_i$ from $N(0, 1000)$, with 5% outliers. The regularization parameter $\gamma$ was chosen through 10-fold cross validation on the training set. Note that in real regression problems one needs to obtain an estimate for the the scale, given by the true standard deviation of the noise in the data. Here we estimated the scale using the mean absolute deviation, a robust approach commonly used in the robust statistics literature [17]. In Table 2, one can see that on both data sets, ClipRelax clearly outperformed the other methods. L2 is clearly skewed by the outliers. Unsurprisingly, the classical robust loss functions, L1 and HuberM, perform better than L2 in the presence of outliers, but not as well as ClipRelax.

**Classification:** We investigated the well known case study from [16] and compared the proposed method to logistic regression (i.e. the logit, or binomial deviance loss [12]) and the robust $1 - \tanh$ loss [19] in a classification context. Here 200 examples were drawn from the target distribution with label noise applied at various levels. The experiment was repeated 50 times to obtain average results and standard deviations. Table 3 shows the test error performance in clean data of the different methods. From these results one can conclude that ClipRelax is more robust than standard logit training. Training with logit loss is slightly better than the $\tanh$ loss algorithm in terms of training loss, but not very significantly. It is interesting to see that when the prediction error is measured on clean labels ClipRelax generalizes significantly better than the robust $1 - \tanh$ loss. This implies that the classification model produced by ClipRelax is closer to the true model despite the presence of outliers, demonstrating that the proposed method can be robust in a simple classification context.

### 7 Conclusion

We have proposed a robust estimation method for regression and classification based on a notion of “loss-clipping”. Although the method is not as fast as standard convex training, it is scalable to problems of moderate size. The key benefit is competitive (or better) estimation quality than the state-of-the-art in robust estimation, while ensuring provable robustness to outliers and computable bounds on the optimality gap. To the best of our knowledge these two properties have never been previously achieved simultaneously. It would be interesting to investigate whether the techniques developed can also be applied to other forms of robust estimators from the classical literature, including GM, MM, L, R and S estimators [11, 13, 17, 23]. Connections with algorithmic stability [3] and influence function based analysis [5, 6, 11] merit further investigation. Obtaining tighter bounds on approximation quality that would enable a proof of consistency also remains an important challenge.

---

### Table 2: Error rates (average root mean squared error, ± standard deviations) for the different regression estimators on various data sets. The values of $(m, n, tt)$ are indicated for each data set, where $m$ is the number of features, $n$ is the number of training examples, and $tt$ the number of testing samples.

<table>
<thead>
<tr>
<th>Loss</th>
<th>Astronomy</th>
<th>Cal-housing</th>
<th>Pumadyn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$(1, 46, 46)$</td>
<td>$(8, 100, 1000)$</td>
<td>$(32, 500, 1000)$</td>
</tr>
<tr>
<td>L2</td>
<td>2.484</td>
<td>804.5 ± 892.5</td>
<td>1.300 × 10^5 ± 68.29</td>
</tr>
<tr>
<td>L1</td>
<td>0.170</td>
<td>0.325 ± 0.046</td>
<td>5.133 ± 0.050</td>
</tr>
<tr>
<td>HuberM</td>
<td>0.149</td>
<td>0.306 ± 0.050</td>
<td>5.377 ± 0.007</td>
</tr>
<tr>
<td>GM (local)</td>
<td>0.166</td>
<td>0.329 ± 0.048</td>
<td>4.399 ± 0.003</td>
</tr>
<tr>
<td>ClipAlt (local)</td>
<td>0.176</td>
<td>0.329 ± 0.048</td>
<td>4.075 ± 1 × 10^{-6}</td>
</tr>
<tr>
<td>ClipRelax</td>
<td>0.131</td>
<td>0.136 ± 0.155</td>
<td>4.075 ± 1 × 10^{-6}</td>
</tr>
</tbody>
</table>

### Table 3: Misclassification error rates on clean data (average error, ± standard deviations) on the Long-Servedio problem [16] with increasing noise levels $p$.

<table>
<thead>
<tr>
<th>Loss</th>
<th>$p = 0.02$</th>
<th>$p = 0.05$</th>
<th>$p = 0.1$</th>
<th>$p = 0.2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logit</td>
<td>4.88 ± 6.17</td>
<td>1.61 ± 7.23</td>
<td>17.67 ± 4.00</td>
<td>19.53 ± 2.91</td>
</tr>
<tr>
<td>L1-tanh (local)</td>
<td>0.91 ± 1.93</td>
<td>2.30 ± 2.85</td>
<td>6.49 ± 4.32</td>
<td>13.96 ± 3.38</td>
</tr>
<tr>
<td>ClipAlt (local)</td>
<td>0.46 ± 0.64</td>
<td>1.51 ± 1.45</td>
<td>4.27 ± 2.57</td>
<td>11.32 ± 3.48</td>
</tr>
<tr>
<td>ClipRelax</td>
<td>0.26 ± 0.34</td>
<td>0.78 ± 0.78</td>
<td>2.49 ± 3.38</td>
<td>10.10 ± 8.21</td>
</tr>
</tbody>
</table>
References


