A User-Friendly Computational Framework for Robust Structured Regression with the \( L_2 \) Criterion

Jocelyn T. Chi\(^a\) and Eric C. Chi\(^b\)

\(^a\)Mathematics, UCLA, Los Angeles, CA; \(^b\)Statistics, Rice University, Houston, TX

ABSTRACT
We introduce a user-friendly computational framework for implementing robust versions of a wide variety of structured regression methods with the \( L_2 \) criterion. In addition to introducing an algorithm for performing \( L_2 \)E regression, our framework enables robust regression with the \( L_2 \) criterion for additional structural constraints, works without requiring complex tuning procedures on the precision parameter, can be used to identify heterogeneous subpopulations, and can incorporate readily available nonrobust structured regression solvers. We provide convergence guarantees for the framework and demonstrate its flexibility with some examples. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received October 2020
Revised January 2021

KEYWORDS
Block-relaxation; Convex optimization; Minimum distance estimation; Regularization

1. Introduction
Linear multiple regression is a classic method that is ubiquitous across numerous domains. Its ability to accurately quantify a linear relationship between a response vector \( y \in \mathbb{R} \) and a set of predictor variables \( X \in \mathbb{R}^{n \times p} \), however, is diminished in the presence of outliers. The \( L_2 \)E method (Terrell 1990; Hjort 1994; Scott 2001, 2009) presents an approach to robust linear regression that optimizes the well-known \( L_2 \) criterion from nonparametric density estimation in lieu of the maximum likelihood. Usage of the \( L_2 \)E method for structured regression problems, however, has been limited by the lack of a simple computational framework. We introduce a general computational framework for performing a wide variety of robust structured regression methods with the \( L_2 \) criterion. Our work offers the following novel contributions.

1. Our framework extends the \( L_2 \)E method from Scott (2001, 2009) to a wide variety of robust structured regression methods with the \( L_2 \) criterion.
2. Our framework enables simultaneous estimation of the regression coefficients and precision parameter (Section 3). We accomplish this via a block-coordinate descent algorithm. Therefore, our simultaneous estimation simplifies the process of choosing a parameter that tunes the robustness of the estimation procedure.
3. Our framework can “robustify” existing implementations of nonrobust structured regression methods in a “plug-and-play” manner (Sections 3.3 and 4).
4. Our framework comes with convergence guarantees for the iterate sequence (Proposition 2).

Section 2 presents motivation for \( L_2 \) robust linear regression. Section 3 introduces our computational framework with convergence guarantees. Section 4 demonstrates the simplicity and flexibility of our framework with robust implementations of several MLE-based methods via existing structured regression solvers. Section 5 provides a discussion.

1.1. Related Work
The \( L_2 \) minimization criterion has been employed in histogram bandwidth selection and kernel density estimators (Scott 1992). Applying this well-known criterion from nonparametric density estimation to parametric estimation for regression problems enables a tradeoff between efficiency and robustness. In fact, Basu et al. (1998) introduced a family of divergences that includes the \( L_2 \)E as a special case and the MLE as a limiting case. The members of this family of divergences are indexed by a parameter that explicitly trades off efficiency for robustness. Meanwhile, the \( L_2 \)E offers a reasonable tradeoff between efficiency and robustness (Warwick and Jones 2005). The robustness of the \( L_2 \)E can also be anticipated since it is a minimum distance estimator, which is known for robustness (Donoho and Liu 1988).

Minimization of the \( L_2 \) criterion has been employed in developing robust statistical models including quantile regression (Lane 2012), mixture models (Lee 2010), classification (Chi and Scott 2014), forecast aggregation (Ramos 2014), and survival analysis (Yang and Scott 2013). It also has uses in engineering applications including signal processing tasks such as wavelet-based image denoising (Scott 2006) and image registration (Ma et al. 2013, 2015; Yang et al. 2017).
Some of the example methods we use to demonstrate our framework in Section 4 have robust implementations. These include robust multiple linear regression (Andrews 1974; Holland and Welsch 1977; Davies 1993; Audibert and Catoni 2011; Meng and Mahoney 2013), robust convex regression (Blanchet et al. 2019), robust isotonic regression (Álvarez and Yohai 2012; Lim 2018), and robust sparse regression (She and Owen 2011; Nguyen and Tran 2013; Alfons, Croux, and Gelper 2013; Ma et al. 2015; Yang, Lozano, and Aravkin 2018; Chang, Roberts, and Welsh 2018). The purpose of our experiments is not to compare the L2E to each of these robust methods. Rather, it is to demonstrate the flexibility and wide applicability of this computational framework and to show how it can obtain robust versions of existing nonrobust implementations in lieu of case-by-case development of robust implementations.

Our framework’s ability to simultaneously optimize over both the precision parameter and regression coefficients is a unique contribution to the literature. To highlight this, we briefly discuss two lines of prior work that are closely related to our proposed framework.

1.1.1. Minimum Distance Estimators for Sparse Regression and Image Registration

In the context of sparse regression, Wang et al. (2013) and Lozano, Meinshausen, and Yang (2016) propose minimum distance estimators that coincide with our formulation when using an $\ell_1$-norm sparsity promoting regularizer. Lozano, Meinshausen, and Yang (2016) employ a modification that applies a log transform on the empirical minimum distance criterion. The key difference between these prior approaches and our framework lies in obtaining the precision parameter. Wang et al. (2013) propose a hybrid block alternating scheme that estimates the regression coefficients by minimizing the L2E criterion with the precision parameter fixed, and then estimates the regression coefficients by solving an optimization problem. This is a modest computational tradeoff compared with searching over a grid of precision parameters.

In the context of image registration, Ma et al. (2013, 2015) and Yang et al. (2017) employ minimum distance estimation to robustly fit a linear model. The primary difference between their work and ours also lies in obtaining the precision parameter. They employ a deterministic annealing approach for choosing the precision parameter. Their algorithm solves an optimization problem to minimize the L2E criterion with respect to the regression coefficients for a fixed precision parameter, and then decreases the precision parameter by a user-defined amount. Finally, they re-estimate the regression coefficients and alternate between updating the regression coefficient estimates and the precision parameter. Once again, a key question is whether the algorithm iterate sequence is guaranteed to converge.

1.1.2. Trimmed Estimators for High-Dimensional Regression

An alternative approach to obtaining robustness is to maximize a trimmed likelihood. Alfons, Croux, and Gelper (2013) employ this for sparse robust multiple linear regression and estimate a sparse regression coefficient vector $\hat{\beta}$ by solving

$$\min_{\beta} \frac{1}{2} \sum_{i=1}^{h} r_{i\ell}(\beta)^2 + \lambda \|\beta\|_1,$$

where $r_i(\beta) = y - X\beta$ is a vector of residuals and $r_{i\ell}(\beta)$ is the $i$th order statistic of $r(\beta)$. The nonnegative parameter $\lambda$ trades off model fit with sparsity in $\beta$. The trimming hyper-parameter $h$ imparts robustness to the standard residual sum of squares term by “trimming away” observations with large residuals. Yang, Lozano, and Aravkin (2018) extend Alfons, Croux, and Gelper (2013) to a general framework for robust penalized estimation similar to ours in the sense that they introduce a single framework for computing structured robust regression problems. The robustness of the estimator hinges on an appropriate $h$. Alfons, Croux, and Gelper (2013) recommend employing prior knowledge to set $h$ while Yang, Lozano, and Aravkin (2018) use cross-validation.

The hyper parameter $h$ plays the same role as the precision parameter in the L2E formulation. A first key difference between the approach in Yang, Lozano, and Aravkin (2018) and ours is that we jointly estimate both the structured model and amount of trimming. This has three benefits. First, we reduce the potential for cross-validation to regularization parameters associated with the structure-incentivizing penalties, for example, $\lambda$ in (1). Second, our framework enables a continuous (and therefore, larger) search space for choosing the precision parameter, as opposed to prespecifying a finite but potentially very large grid of trimming parameters for many observations. Third, our framework estimates both the regression coefficients and the precision parameter within an optimization framework, enabling convergence guarantees over the iterates.

A second key difference between the approach in Yang, Lozano, and Aravkin (2018) and ours is that the precision parameter in our framework performs a “soft-trimming” action by adaptively choosing new down-weights for observations that
are less consistent with the proposed model in each iteration. Rather than a single trim applied to all the observations, this enables additional flexibility for automatically varying the contribution of individual observations to the model fit. Section 4.4 demonstrates these advantages.

2. Robust Regression with the L2 Criterion

Let \( f \) be the true but unknown density generating the observed data \( y_1, \ldots, y_n \in \mathbb{R} \), and let \( \hat{f}_\theta \) be a probability density function indexed by a parameter \( \theta \in \Theta \subset \mathbb{R}^d \) approximating \( f \). We assume throughout that all vectors are column vectors. If we were to estimate \( f \) using the \( \hat{f}_\theta \) that is closest to it, we could minimize the L2 distance between \( f \) and \( \hat{f}_\theta \) in lieu of the negative log-likelihood with

\[
\min_{\theta \in \Theta} \int \left[ \hat{f}_\theta(y) - f(y) \right]^2 dy. \tag{2}
\]

In practice, however, identifying \( \hat{\theta} \) in this way is impossible since \( f \) remains unknown. While we typically cannot minimize the L2 distance between \( f \) and its estimate \( \hat{f}_\theta \) directly, we can minimize an unbiased estimate of this distance. To do this, we first expand (2) as

\[
\int \hat{f}_\theta(y)^2 dy - 2 \int \hat{f}_\theta(y) f(y) dy + \int f(y)^2 dy.
\]

Notice that the second integral is the expectation \( E_f(\hat{f}_\theta(Y)) \), where \( Y \) is a random variable drawn from \( f \). Therefore, the sample mean provides an unbiased estimate of this quantity. Meanwhile, the third integral does not depend on \( \theta \). Therefore, we arrive at the following fully data-based loss function \( h(\theta) \) that provides an unbiased estimate for (2) up to an irrelevant additive constant

\[
h(\theta) = \int \hat{f}_\theta(y)^2 dy - \frac{2}{n} \sum_{i=1}^{n} \hat{f}_\theta(y_i), \tag{3}
\]

assuming \( \hat{f} \) is square integrable over an appropriate region. Minimizing over this fully observed loss function presents us with our estimator \( \hat{\theta} \), also called an L2E (Scott 2001). Section 3.2 provides intuition as to how the L2E imparts robustness in our framework.

2.1. Regression Model Formulation

Let \( y \in \mathbb{R} \) denote a vector of \( n \) observed responses and let \( X \in \mathbb{R}^{n \times p} \) denote the corresponding observed design matrix of \( p \)-dimensional covariates. The standard linear model assumes the response and covariates are related via the model

\[
y = X\beta_0 + \tau_0^{-1} \varepsilon,
\]

where \( \beta_0 \in \mathbb{R}^p \) is an unobserved vector of regression coefficients, \( \tau_0 \in \mathbb{R}_+ \) is an unobserved precision parameter, and the unobserved noise \( \varepsilon_i \in \mathbb{R} \) for \( 1 \leq i \leq n \) are iid standard Gaussian random variables. We phrase the regression model in terms of the precision rather than the variance to obtain a more straightforward optimization problem later.

Let \( \theta = (\beta^T, \tau)^T \) denote the vector of unknown parameters. Additionally, let \( r \) denote the residual vector obtained from the current prediction estimate for \( \beta \) so that its \( i \)th component is \( r_i = y_i - x_i^T \beta \), where \( x_i \in \mathbb{R}^p \) contains the \( i \)th row of \( X \). Given any suitable pair of \( \beta \) and \( \tau \), the conditional density of \( y_i \) for \( 1 \leq i \leq n \) is

\[
\hat{f}_\theta(y_i) = \frac{\tau}{\sqrt{2\pi}} \exp\left( -\frac{\tau^2 y_i^2}{2} \right).
\]

Following Scott (2001), we use the L2E loss function for linear regression by averaging the L2 distance over the observed data and minimize

\[
h(\theta) = \frac{1}{n} \sum_{i=1}^{n} h(\theta_i) = \frac{1}{2\sqrt{\pi}} - \frac{\tau}{n} \sqrt{2\pi} \sum_{i=1}^{n} \exp\left(-\frac{\tau^2 y_i^2}{2} \right), \tag{4}
\]

where

\[
h(\theta_i) = \int_{-\infty}^{\infty} \left[ \hat{f}_\theta(y_i) \right]^2 dy_i - 2 \hat{f}_\theta(y_i)
\]

\[
= \frac{\tau}{2\sqrt{\pi}} - \tau \sqrt{2\pi} \exp\left(-\frac{\tau^2 y_i^2}{2} \right).
\]

The solution \( \hat{\theta} = (\hat{\beta}^T, \hat{\tau})^T \) of (4) contains the L2E regression estimates.

3. Computational Framework

We pose our estimation and model fitting task as a nonsmooth optimization problem. For references on optimization techniques employed in this article, please refer to Lange, Chi, and Zhou (2014), Polson, Scott, and Willard (2015) and Lange (2013, 2010). Our computational framework for performing robust structured regression via the L2 criterion is a general algorithm that combines the L2E method (Scott 2001, 2009) with a structural constraint or penalty term \( \phi(\beta) \). As an example, suppose we wish to enforce a nonnegativity constraint on the regression coefficients \( \beta \). Then we can take \( \phi(\beta) = \nu_C(\beta) \), the indicator function of the nonnegative orthant \( C = \{ \beta \in \mathbb{R}^p : \beta_j \geq 0, 1 \leq j \leq p \} \). Recall that the indicator function of a set \( C \), denoted \( \nu_C(\beta) \), is a function that takes values on the extended reals and is zero when \( \beta \in C \) and is \( \infty \) otherwise. As another example, \( \phi(\beta) \) may be an indicator function requiring that the elements of \( \beta \) satisfy a monotonicity constraint. Other examples include taking \( \phi(\beta) \) to be sparsity inducing penalties like the \( \ell_1 \)-norm (Tibshirani 1996) or elastic net (Zou and Hastie 2005). Section 4 contains several examples of potential constraint terms \( \phi(\beta) \). Concretely, we seek a minimizer of the objective function

\[
\ell(\beta, \tau) = h(\beta, \tau) + \phi(\beta) \tag{5}
\]

subject to \( \beta \in \mathbb{R}^p \) and \( \tau \in [\tau_{\text{min}}, \tau_{\text{max}}] \), where \( \tau_{\text{min}} \in \mathbb{R} \) and \( \tau_{\text{max}} \in \mathbb{R} \) are minimum and maximum values for \( \tau \), respectively.

There are two computational challenges in minimizing (5). The first is that \( \ell \) is nonconvex in \( \theta \) since \( h(\theta) \) is nonconvex. The second is that commonly used constraint terms \( \phi(\beta) \) are often nonsmooth or nondifferentiable. We focus on the case where the \( \phi \) are nonnegative, continuous, and convex functions.
Continuity and convexity ensure that the proximal mappings of \( \phi \) always exist and are unique. In minimizing (5), we use the key property that the block derivatives of \( h \) with respect to \( \beta \) and \( \tau \), that is \( \nabla_\beta h(\beta, \tau) \) and \( \frac{\partial}{\partial \tau} h(\beta, \tau) \), respectively, are Lipschitz differentiable.

**Proposition 1.** The \( L^2_\text{E} \) loss function \( h(\beta, \tau) \) is block Lipschitz differentiable with respect to \( \beta \) and \( \tau \) so that

\[
\| \nabla_\beta h(\beta, \tau) - \nabla_\beta h(\tilde{\beta}, \tilde{\tau}) \|_2 \leq L_\beta(\tau) \| \beta - \tilde{\beta} \|_2
\]

for all \( \beta \) and \( \tilde{\beta} \), and

\[
\left| \frac{\partial}{\partial \beta} h(\beta, \tau) - \frac{\partial}{\partial \beta} h(\tilde{\beta}, \tilde{\tau}) \right| \leq L_\beta(\tau) |\tau - \tilde{\tau}|
\]

for all \( \tau \) and \( \tilde{\tau} \). The Lipschitz constant \( L_\beta(\tau) \) is \( L_\beta(\tau) = \frac{\tau^2}{2} \sigma(X)^2 \), where \( \sigma(X) \) is the largest singular value of \( X \). The Lipschitz constant \( L_\tau(\beta) \) is \( L_\tau(\beta) = \frac{3}{2\sqrt{\pi}} \rho \exp\left(-\frac{1}{2}\right) \), where \( \rho = \min_{i\neq 0} |\tau_i| \).

The supplementary materials contains the proof. The block Lipschitz differentiability of \( h(\beta, \tau) \) and the regularity conditions on \( \phi \) lead us to employ block coordinate descent to minimize (5). At a high level, we alternate between minimizing with respect to \( \beta \) holding \( \tau \) fixed, and minimizing with respect to \( \tau \) holding \( \beta \) fixed. We solve two subproblems at the \( k \)th update:

**Subproblem 1:** Update \( \beta \)

\[
\beta^{(k)} = \arg \min_{\beta} h(\beta, \tau^{(k-1)}) + \phi(\beta), \quad \text{(6)}
\]

**Subproblem 2:** Update \( \tau \)

\[
\tau^{(k)} = \arg \min_{\tau} h(\beta^{(k)}, \tau). \quad \text{(7)}
\]

In practice, we do not exactly solve either subproblem and instead take a few proximal gradient descent steps to partially minimize or inexacty solve (6) and (7). Note that each update is guaranteed to monotonically decrease the loss function \( \ell(\theta) \). This is a feature that all block coordinate descent algorithms possess as a special case of majorization-minimization algorithms (Lange 2016).

Recall that proximal gradient descent is a first-order iterative method for solving optimization problems of the form

\[
\min_{\theta} h(\theta) + \phi(\theta), \quad \text{(8)}
\]

where \( h \) is a Lipschitz differentiable function and \( \phi \) is a convex and lower semicontinuous function (Combettes and Wajs 2005; Parikh and Boyd 2014). Further recall that the proximal map of \( \phi \)

\[
\text{prox}_\phi(\theta) = \arg \min_{\hat{\theta}} \frac{1}{2} \| \hat{\theta} - \theta \|_2^2 + \phi(\hat{\theta})
\]

exists and is unique whenever \( \phi(\theta) \) is convex and lower semicontinuous. Many regularizers \( \phi(\beta) \) that are useful for recovering models with structure satisfy these conditions and also admit proximal maps that can be evaluated via explicit formulation or an efficient algorithm. For example, the proximal map of the scaled \( \ell_1 \)-norm \( \lambda \| \cdot \|_1 \) is the element-wise soft-thresholding operator, namely

\[
\text{prox}_{\lambda \| \cdot \|_1}(\theta)_i = \text{sign}(\theta_i) \max(|\theta_i| - \lambda, 0). \quad \text{(9)}
\]

Notice that the proximal map can be viewed as the generalization of the Euclidian projection, which we refer to as the projection. Specifically, the projection of a point \( \theta \) onto a set \( C \) is the point \( \text{P}_C(\theta) = \text{C} \) that is closest in Euclidean distance to \( \theta \), namely

\[
\text{P}_C(\theta) = \arg \min_{\theta \in C} \| \theta - \theta \|_2.
\]

Similarly, the proximal map of the indicator function \( i_C \) of a set \( C \) is the projection onto the set \( C \). This projection exists and is unique when \( C \) is a closed convex set. For example, when \( C = [r_{\min}, r_{\max}], \text{prox}_{i_{[r_{\min}, r_{\max}]}}(\tau) = \text{P}_{[r_{\min}, r_{\max}]}(\tau) \). As its name suggests, the proximal gradient descent method for solving problems of the form in (8) combines a gradient descent step with a proximal step. Given an iterate \( \theta \), the update \( \theta^+ \) is

\[
\theta^+ = \text{prox}_{t\phi}(\theta - t\nabla h(\theta)), \quad \text{(10)}
\]

where \( t \) is a positive step-size parameter and \( t\phi \) is the function \( \phi \) scaled by \( t \).

We emphasize that our framework does not require exactly computing the global minimizers in (6) and (7) at each iteration. Nonetheless, we will see that the algorithm still comes with some convergence guarantees.

**Remark.** We make the modestly stronger assumption that \( \phi \) is continuous to establish convergence guarantees. Assuming continuity is not restrictive as commonly employed, convex nonsmooth \( \phi \) includes norms, compositions of norms with linear mappings, and indicator functions of closed convex sets.

### 3.1. A General Algorithm for \( L^2_\text{E} \) Robust Structured Regression

Algorithm 1 presents pseudocode for minimizing (5) via inexact block coordinate descent. For the update step on \( \tau \), the operator \( \text{P}_{[r_{\min}, r_{\max}]} \) denotes the projection onto \( [r_{\min}, r_{\max}] \). When updating \( \beta \) in (6) and \( \tau \) in (7), we take a fixed number of proximal gradient steps, \( N_\beta \) and \( N_\tau \), respectively, in (10). The gradients for updating \( \beta \) and \( \tau \) are

\[
\nabla_\beta h(\beta, \tau) = -\frac{\tau^2}{2n} X^T W r, \quad \text{(11)}
\]

where \( W \in \mathbb{R}^{n \times n} \) is a diagonal matrix that depends on \( \beta \) with \( \beta \)th diagonal entry

\[
w_{\beta i} = \exp\left[-\frac{\tau^2}{2r_i^2}\right], \quad \text{and} \quad \text{(12)}
\]

\[
\frac{\partial}{\partial \tau} h(\beta, \tau) = \frac{1}{2\sqrt{\pi}} - \frac{1}{n\sqrt{\pi}} \sum_{i=1}^{n} w_{\beta i} (1 - \tau^2 r_i^2).
\]

Algorithm 1 has the following convergence guarantee. Recall that a point \( \theta = (\beta^T, \tau^T) \) is a first-order stationary point of a function \( f(\theta) \) if for all directions \( v \), the directional derivative \( f^\prime(\theta; v) \) of \( f \) is nonnegative.
**Proposition 2.** For any choice of $N_β$ and $N_τ$, under modest regularity conditions on (5) and step-sizes $t_β = L_β(τ(k))^{-1}$ and $t_τ = L_τ(τ)$, where $L_β(τ)$ and $L_τ(β)$ are in Proposition 1, the sequence $(β(k), τ(k))$ generated by Algorithm 1 has at least one limit point and all limit points are first-order stationary points of (5). If there are finitely many first-order stationary points of (5), then the sequence $(β(k), τ(k))$ will converge to one of them.

The supplementary materials contains a proof. We briefly comment on the assumption about the number of stationary points. It may seem strong to assume that the $L_2E$ objective function in (6) and the update rule for Algorithm 1. This scenario applies to isotonic and convex regression, where each element $y_i$ of the responses, where each element $y_i$ of the observation resembles the observed response $y_i$.

There are some assumptions on the number of stationary points that the $L_2E$ objective function in (6) and the update rule for Algorithm 1. This scenario applies to isotonic and convex regression, where each element $y_i$ of the responses, where each element $y_i$ of the observation resembles the observed response $y_i$.

**Algorithm 1** Block coordinate descent for minimizing (5)

```
Initialize $β^{(0)}$, $τ^{(0)}$ and fix $N_β, N_τ$
1: $k ← 0$
2: repeat
3: $t_β ← t_β(τ(k))^{-1}$ // Update $L_β(τ(k))$ via Proposition 1
4: $β ← β^{(k)}$ // Update $β$ (6)
5: for $i = 1, ..., N_β$ do
6: $β ← \text{prox}_{t_β φ}(β - \nabla β h(β, τ(k)))$
7: end for
8: $β^{(k+1)} ← β$
9: $τ ← τ(k)$ // Update $τ$ (7)
10: for $i = 1, ..., N_τ$ do
11: $τ ← \text{prox}_{t_τ φ}(τ - \nabla τ φ(β^{(k+1)}, τ))$
12: end for
13: $τ(k+1) ← τ$
14: $k ← k + 1$
15: until convergence
```

### 3.2. Algorithmic Intuition

We present a simple scenario illustrating intuition for Algorithm 1. This scenario applies to isotonic and convex regression, which we discuss in Section 4. Let the design matrix $X$ be the $n \times n$ identity matrix $I_n$, and let the structural constraint $φ(β)$ be the indicator function of a closed and convex set $C$. Then $φ(β) = I_C(β)$ is zero if $β ∈ C$, and is infinite otherwise. This results in simplifications to (6) and the update rule for $β$ becomes

$$β^+ = P_C(β),$$

where $P_C(β)$ is the Euclidean projection of $β$ onto $C$, and $W$ is a diagonal matrix with diagonal elements as defined in (12).

We observe how the $L_2E$ imparts robustness through the action of $W$. Consider $z ∈ \mathbb{R}^n$ as a vector of pseudo-observations, where each element $z_i$ is a convex combination of $y_i$ and the current prediction $β_i$. If the current residual $r_i$ is large compared to the current precision $τ$, $w_i$ is small and the corresponding pseudo-observation $z_i$ resembles the current predicted value $β_i$. Meanwhile, if the current residual $r_i$ is small compared to the current precision $τ$, the corresponding pseudo-observation resembles the observed response $y_i$.

Therefore, the pseudo-observations impart the following algorithmic intuition. Given an estimate $β$ of the regression coefficients, the algorithm performs constrained least squares regression using a pseudo-response $z$, whose entries are a convex combination of the entries of the observed response $y$ and the prediction $β$. Observations with large current residuals relative to the current precision are essentially replaced by their predicted value.

In this way, the algorithm can fit a fraction of the observations very well while also accounting for outlying observations by replacing them with pseudo-response values more consistent with a model that fits the data. Notice that the algorithm is oblivious to whether large residuals arise from outliers in the response or in the predictor variables. Consequently, it can handle outliers arising from either source or both.

### 3.3. Robustifying Existing Nonrobust Implementations

We now discuss how one can employ this framework to automatically “robustify” existing nonrobust structured regression implementations solving problems of the form

$$\min_β \frac{1}{2} ||y - Xβ||_2^2 + φ(β).$$

Concretely, we can use existing nonrobust solvers to perform line 6 in Algorithm 1. Recall that line 6 performs the $β$ update with

$$β^+ = P_C(β) \quad \text{or} \quad β^+ = \text{prox}_{t_β φ}(z)$$

depending on whether $φ$ is a projection operator or a more general proximal mapping. In both cases, we perform this step by calling the existing nonrobust solver and inputting $z$ in place of the original response $y$. Our computation for $z$ depends on whether $X$ is the identity. If $X$ is the identity, as in isotonic and convex regression, then $z$ in Algorithm 1 line 6 simplifies to

$$z = Wγ + (I - W)β$$

in place of $y$ into the existing nonrobust solver.

If $X$ is not the identity, as in Lasso regression, then $z$ in Algorithm 1 line 6 is the more complex

$$z = β - t_β \nabla β h(β, τ)$$

with $\nabla β h(β, τ)$ as described in (11). Recall that the $β$ update involves a penalized least squares problem with identity design $I$ (Section 3.2)

$$β^+ = \text{prox}_{t_β φ}(z) = \minimize_β \frac{1}{2} ||z - Iβ||_2^2 + t_β φ(β).$$

Therefore, Algorithm 1 line 6 inputs $z = β - t_β \nabla β h(β, τ)$ in place of $y$ and $I$ in place of $X$ into the existing nonrobust solver.

### 3.4. Practical Considerations

We discuss guidance on setting the hyperparameters in Algorithm 1. The constraint set $[τ_{min}, τ_{max}]$ on $τ$ was introduced to establish the existence of a limit point for the algorithm iterate sequence. In practice, the constraints do not appear to

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS

1055

15:
strongly influence performance. Nonetheless, it is possible to run into a numerical issue if $\tau_{\min}$ is set to zero. Specifically, it is possible that the gradient step in the $\tau$-update outputs a negative value, which would then be projected to 0. This results in $L_\phi(\tau)$ set to zero, which leads to an undefined step-size $t_\phi$.

To guard against this, we recommend setting $\tau_{\min}$ as follows. A conservative estimate of the standard deviation follows from assuming no association between the response and covariates and all the variation in the response $y$ is due to noise, namely

$$\hat{\sigma} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2}.$$ 

Therefore, take $\tau_{\min} = \hat{\sigma}^{-1}$. For the upper bound, taking $\tau_{\max}$ to be infinity does not appear to create any issues in practice.

A natural question is how to set $N_\beta$ and $N_\tau$ in Algorithm 1. Choosing these values too small or too large can lead to slow convergence. In our experience, setting $N_\beta = N_\tau = 1$ does not make sufficient progress in minimizing the objective functions in (6) and (7). Meanwhile, setting $N_\beta$ and $N_\tau$ to be a larger value such as 1000 leads to diminishing returns in minimizing the objective functions (6) and (7). In our experiments, we set $N_\beta$ and $N_\tau$ to be 100 as it strikes a balance between these two extremes.

Finally, given the nonconvexity of the $L_2E$ objective function in (5), some thought to initialization is required. We recommend the following simple “null model” initialization strategy. When we have a nonidentity design matrix $X$, similar to choosing $\tau_{\min}$, we assume there is no association between the response and covariates. Therefore, we set the initial regression coefficient vector $\beta^{(0)} = 0$. When $X$ is the identity, we set $\beta^{(0)} = \bar{y}1$, namely the vector of all ones multiplied by the mean $\bar{y}$ of the response $y$. In both cases, we set the initial precision to be $\tau^{(0)} = \text{MAD}(y)^{-1}$, the reciprocal of the median absolute deviations of the response $y$. We employ this initialization strategy throughout Section 4. The supplementary materials contains a simulation study demonstrating that the output of Algorithm 1 appears stable to perturbations in this initialization heuristic.

### 4. Examples of L2E Robust Structured Regression

We demonstrate our framework on a variety of robust structured regression methods. Our examples illustrate how our framework can “robustify” existing structural regression solvers. We begin with multivariate L2E regression (Scott 2001, 2009), where $\phi(\beta) = 0$. Let $X \in \mathbb{R}^{n \times p}$ with $\text{rank}(X) = p$. The data come from an Italian bank (Riani et al. 2014) where the response $y \in \mathbb{R}$ is the annual investment earnings for $n = 1949$ banking customers. The design $X$ contains measurements on $p = 13$ bank services.

Since $\phi(\beta) = 0$, prox$_\beta$ is simply the identity operation. Subproblem 1 for updating $\beta$ in (6) reduces to iteratively performing: (a) Compute current residuals, (b) Update weights $w_{\text{ii}}$ in (12), and (c) Update $\beta$ with current residuals and gradient described in Section 3.1.

Figures 1(a),(b) depict scatterplots of the fitted values against the residuals. A good fit is evidenced by normally distributed noise in the residuals, or symmetric scatter of points about the zero residual level. Figure 1(a) shows a discernible pattern in the MLE residuals with asymmetric scatter of points about the zero residual level. This indicates that additional trends in the data not captured by the Gaussian linear model remain in the residuals and are not captured by the MLE fit.

Meanwhile, Figure 1(b) shows that after excluding outlying points identified by automatic tuning of $\tau$ in our framework (depicted by the blue triangles), the residuals from the L2E fit are normally distributed about zero. To identify outliers, we compute the L2E residuals and select observations whose residuals exceed a factor of the precision parameter, for example, 3 divided by $\tau$. Thus, the L2E adequately captures the linear relationship between investment earnings and bank services for the nonoutlying customers. Notice that one can recursively repeat L2E regression on outlying customers to identify an appropriate linear relationship between investment earnings and bank services for customer subgroups.

This example also highlights how our framework enables joint estimation of the regression coefficient vector $\beta$ and the precision $\tau$, enabling automatic identification of outlying observations in the data. This is practically useful since the L2E can simultaneously identify subpopulations within the data and appropriate fits for each of those groups when applied recursively to the subgroups.

#### 4.2. L2E Robust Isotonic Regression

Our next example is L2E robust isotonic regression. Let an observed response $y \in \mathbb{R}^n$ consist of $n$ samples drawn from a monotonic function $f$ sampled at discrete time points $t_1 \leq t_2 \leq \cdots \leq t_n$ with additive independent Gaussian noise. The $i$th entry of $y$ is

$$y_i = f(t_i) + \epsilon_i \quad \text{for} \quad 1 \leq i \leq n,$$

where $f$ is monotonic, $\epsilon_i \overset{\text{iid}}{\sim} N(0, \frac{1}{\tau})$, and $\tau \in \mathbb{R}_+$. The goal of isotonic regression (Brunk et al. 1972; Barlow and Brunk 1972; Lee 1981; Dykstra and Robertson 1982; Mair, Hornik, and de Leeuw 2009) is to estimate $f$ by solving

$$\min_{\beta(t_1), \ldots, \beta(t_n)} \sum_{i=1}^{n} (y_i - \beta(t_i))^2$$

subject to $\beta(t_1) \leq \beta(t_2) \leq \cdots \leq \beta(t_n)$.

We construct a piece-wise constant estimate for $f$ using the elements of the estimator $\hat{\beta} = (\hat{\beta}(t_1) \quad \hat{\beta}(t_2) \quad \cdots \quad \hat{\beta}(t_n))^T$. For the corresponding L2E problem, the design $X = I_n$ and $\phi(\beta) = I_M(\beta)$ is the indicator function over the set of vectors $M$ satisfying element-wise monotonicity so that $\beta_1 \leq \beta_2 \leq \cdots \leq \beta_n$ for $\beta \in \mathbb{R}^n$. Subproblem 1 for updating $\beta$ in (6) reduces to iteratively performing: (a) Compute current residuals, (b) Update weights $w_{\text{ii}}$ in (12), and (c) Update $\beta$ with current residuals and gradient described in Section 3.1 and...
Meanwhile, Figure 2(b) shows how the MLE is skewed from generalized PAVA while the solid blue line depicts the true underlying cubic fit. The dashed orange line depicts the MLE and the L2E while varying the number of outliers.

To create high leverage points in the covariates space, we employ the gpava function in the isotone package (Mair, Hornik, and de Leeuw 2009) for R to obtain the MLE. We obtain 100 replicates for each scenario on a 3.00 GHz Intel Core i7 computer with 32 GB of RAM and present boxplots of the mean squared error (MSE) and time in seconds. We obtain the MSE between the model $y$ and the computed solution.

The MLE produces increasingly larger MSE as the number of outliers increases. Meanwhile, the L2E produces a smaller increase in MSE for the same number of outliers but requires more computation time since the L2E employs multiple computations of the MLE procedure. Thus, the L2E can produce an isotonic regression fit that is much less sensitive to outliers than the MLE.

### 4.3. L2E Robust Convex Regression

Our next example is L2E robust convex regression. For illustration, we consider the univariate case (Wang and Ghosh 2012; Ghosal and Sen 2017). However, our framework applies to multivariate convex regression (Meyer 2003; Birke and Dette 2007; Seijo and Sen 2011; Hannan and Glynn 2012; Guntuboyina and Sen 2015; Aybat and Wang 2016; Mazumder et al. 2019; Lin, Sun, and Toh 2020; Bertsimas and Mundru 2021; Chen and Mazumder 2021) in a similar manner. Let an observed response $y_i$ drawn from a convex function $f$ sampled at discrete time points $t_1 \leq t_2 \leq \cdots \leq t_n$ with additive independent Gaussian noise. The $i$th entry of $y$ is

$$ y_i = f(t_i) + \epsilon_i \quad \text{for} \quad 1 \leq i \leq n, $$

for convex $f$, $\epsilon_i \overset{iid}{\sim} N(0, \frac{1}{\tau})$, and $\tau \in \mathbb{R}^+$. Shape-restricted convex regression estimates $f$ via

$$ \min_{\beta(t_1), \ldots, \beta(t_n)} \sum_{i=1}^{n} (y_i - \beta(t_i))^2 $$

subject to $\beta(t_i) \leq \beta(t_{i+1}) - \frac{t_{i+1} - t_i}{t_{i+1} - t_{i-1}} \beta(t_{i-1}) + \frac{t_i - t_{i-1}}{t_{i+1} - t_{i-1}} \beta(t_{i+1})$ for $2 \leq i \leq n - 1$.

Let $\hat{\beta} = (\hat{\beta}(t_1) \quad \hat{\beta}(t_2) \quad \cdots \quad \hat{\beta}(t_n))^T$. We recast this constraint in terms of a scaled second-order differencing matrix $D \in \mathbb{R}^{n \times n}$ with $D \hat{\beta} \succeq 0$ so that all the elements of $D \hat{\beta}$ are nonnegative. We construct a piece-wise constant estimate for $f$ using the elements of $\hat{\beta}$.
For the corresponding $L_2E$ problem, the design $X = I_n$ and $\phi(\beta) = l_C(\beta)$ is the indicator function over the set of vectors in $C = \{ \beta : D\beta \geq 0 \}$. Subproblem 1 for updating $\beta$ in (6) reduces to iteratively performing: (a) Compute current residuals, (b) Update weights $w_{ij}$ in (12), and (c) Update $\beta$ with current residuals and gradient described in Section 3.1 and project onto the convex cone $C$. The \texttt{conreg} function in the \texttt{cobs} package (Ng and Maechler 2007) for R performs this last step so we can employ it in Algorithm 1 line 6.

Figure 4(a) shows how the MLE and $L_2E$ produce similar fits in the absence of outliers. The true underlying convex fit $f$ in black and gray points depict observations generated from $f$ with additive Gaussian noise. The dashed orange line depicts the MLE obtained from the \texttt{cobs} package in R while the solid blue line depicts the $L_2E$. Meanwhile, Figure 4(b) shows how the MLE is substantially skewed toward the outliers while the $L_2E$ is less distorted. This example highlights how the $L_2E$ is less sensitive to outliers than the MLE.

Figure 5 depicts results of Monte Carlo simulations comparing the MLE and $L_2E$ on shape-restricted convex regression while varying the number of outliers. We simulate three datasets with $n = 1000$ observations using a fourth-order polynomial...
with additive Gaussian noise and 50, 100, and 200 outliers, respectively. We introduce outliers by selecting points from approximately the 25th quartile along the x-axis and assigning them a value that is equal to a little less than the maximal polynomial value and additive standard Gaussian noise. This corresponds to simulating samples from a bimodal distribution to create high leverage points in the covariate space. We employ the conreg function in the cobs package for R (Ng and Maechler 2007) to obtain the MLE. We obtain 100 replicates on a 3.00 GHz Intel Core i7 computer with 32 GB of RAM.

Figure 5(a) highlights how the MLE produces increasingly larger MSE values as the number of outliers increases. Meanwhile, the $L_2E$ MSE is much less sensitive to outliers. This example again underscores how our framework can perform a robust version of a structured regression problem using a readily available nonrobust implementation.

4.4. $L_2E$ Robust $\ell_1$ Penalized Regression

Our last example is $L_2E$ $\ell_1$ penalized regression. We use the Lasso (Tibshirani 1996)
as our reference. For the corresponding L2E problem, let \( X \in \mathbb{R}^{n \times p} \) with rank(\( X \)) = \( p \) and let \( \phi(\beta) = \lambda \| \beta \|_1 \). Subproblem 1 for updating \( \beta \) in (6) reduces to iteratively performing: (a) Compute current residuals, (b) Update weights \( w_i \) in (12), and (c) Update \( \beta \) with current residuals and gradient in Section 3.1 and apply the soft-thresholding operator in (9).

We illustrate with real data on prostate cancer patients from Stamey et al. (1989). The response \( y \in \mathbb{R}^n \) is the percent of Gleason score (measure of a prostate-specific antigen) for \( n = 97 \) patients receiving a radical prostatectomy. The design \( X \) contains measurements on \( p = 8 \) clinical variables. To introduce outliers in the covariates, we identify the top 5% of observations in \( X \) with highest leverage and scale these points by 3.3.

Figure 7 in the supplementary materials depicts solution paths for Lasso, L2E \( \ell_1 \) penalized regression, sparse least trimmed squares (Sparse LTS) (Alfons, Croux, and Gelper 2013; Yang, Lozano, and Aravkin 2018), and exponential squared loss Lasso (ESL Lasso) (Wang et al. 2013) as a function of the shrinkage factor \( s = \| \hat{\beta}(\lambda) \|_1 \). We set \( \hat{\beta}_0 \) as the \( \beta \) estimate obtained at \( \lambda = 0 \) for each method and employ a \( \lambda \) sequence with a log-linear scale of 15 values between \( 10^{-5} \) and a conservative data-dependent estimate of \( \lambda \) at which \( \hat{\beta}(\lambda) = 0 \).

Since all methods employ the \( \ell_1 \) penalty, the latter three can be viewed as alternative approaches to robust Lasso. Therefore, the Lasso solution paths, which quantify the relative contributions of the covariates to the regression model, without outliers (top-left panel) serve as a control. Ideally, a robust implementation preserves these relative contributions in the presence of outliers. Qualitatively, the L2E solution paths most closely resemble the Lasso solution paths and suffer the least distortion in the presence of outliers. By comparison, Sparse LTS and ESL Lasso qualitatively appear very different from Lasso, even without outliers. We employ the default trimming percentage (retains 75% of the data) for Sparse LTS so it should be robust to the 5% of outliers.

Section 5.1 of the supplementary materials contains more quantitative experiments with these four methods using synthetic data. Table 3 of the supplementary materials shows that the L2E obtains lower relative error on average and additionally selects fewer false positives. Although Sparse LTS and ESL Lasso employ the \( \ell_1 \) penalty for variable selection, they both select nearly all the variables in those experiments in the presence of outliers.

5. Discussion

Least squares regression models can be extended to encode a wide array of prior structure through nonsmooth penalties and constraints. While regression via least squares—and its constrained and penalized extensions—does not require any parametric assumptions, making a normality assumption on the residuals opens the door to applying the L2E method for robustly fitting a parametric regression model. In this work, we introduce a user-friendly computational framework, or recipe, for performing a wide variety of robust structured regression methods by minimizing the L2 criterion. We highlight that our framework can “robustify” existing structured regression solvers by using existing nonrobust solvers in the \( \beta \)-update step in a plug-and-play manner. Thus, our framework can readily incorporate newer and improved technologies for existing structured regression methods; as faster and better algorithms for these nonrobust structured regression solvers appear, users may simply replace the previous solver with the new one in the \( \beta \)-update step.

We also highlight the significance of the convergence properties of our computational framework. As long as the structural constraints or penalties satisfy convexity and continuity conditions, a solution obtained with our framework is guaranteed to converge to a first-order stationary point. Since many commonly-used structural constraints and penalties satisfy these conditions, our framework provides convergence guarantees for robust versions of many nonrobust methods with readily available software.

We close by noting that our L2E framework focuses on structured regression problems under a normality assumption, which may not be appropriate in all situations. Meanwhile, the L2E framework has also been used to robustly estimate parametric models under different distributional assumptions, for example, Weibull (Yang and Scott 2013), Poisson (Scott 2001), and logistic (Chi and Scott 2014). An interesting direction for future work is the development of a unified computational framework for fitting structured regression models under a wider range of distributional assumptions.

Supplementary Materials

Title: Supplement to “A User-Friendly Computational Framework for Robust Structured Regression with the L2 Criterion” (tex file)

Software: L2E R-package for performing L2E structured regression. (GNU ziped tar file)

Funding

The work was supported in part by NSF grants DMS-2201136 and DMS-2103093, and NIH grant R01GM135928.

ORCID

Jocelyn T. Chi http://orcid.org/0000-0001-6503-3661
Eric C. Chi http://orcid.org/0000-0003-4647-0895

References

Alfons, A., Croux, C., and Gelper, S. (2013), “Sparse Least Trimmed Squares Regression for Analyzing High-Dimensional Large Data Sets,” *The Annals of Applied Statistics*, 7, 226–248. [1052, 1060]
Álvarez, E. E., and Yohai, V. J. (2012), “M-estimators for Isotonic Regression,” *Journal of Statistical Planning and Inference*, 142, 2351–2368. [1052]
Andrews, D. F. (1974), “A Robust Method for Multiple Linear Regression,” *Technometrics*, 16, 523–531. [1052]
Audibert, J.-Y., and Catoni, O. (2011), “Robust Linear Least Squares Regression,” *The Annals of Statistics*, 39, 2766–2794. [1052]
Aybat, N. S., and Wang, Z. (2016), “A Parallelizable Dual Smoothing Method for Large Scale Convex Regression Problems,” arXiv:1608.02227 [math.OC]. [1057]
Wang, X., Jiang, Y., Huang, M., and Zhang, H. (2013), “Robust Variable Selection with Exponential Squared Loss,” *Journal of the American Statistical Association*, 108, 632–643. [1052,1060]

Warwick, J., and Jones, M. (2005), “Choosing a Robustness Tuning Parameter,” *Journal of Statistical Computation and Simulation*, 75, 581–588. [1051]

Yang, E., Lozano, A. C., and Aravkin, A. (2018), “A General Family of Trimmed Estimators for Robust High-Dimensional Data Analysis,” *Electronic Journal of Statistics*, 12, 3519–3553. [1052,1060]

Yang, J., and Scott, D. W. (2013), “Robust Fitting of a Weibull Model with Optional Censoring,” *Computational Statistics & Data Analysis*, 67, 149–161. [1052,1060]

Yang, K., Pan, A., Yang, Y., Zhang, S., Ong, S. H., and Tang, H. (2017), “Remote Sensing Image Registration Using Multiple Image Features,” *Remote Sensing*, 9, 581. [1051,1052]

Zou, H., and Hastie, T. (2005), “Regularization and Variable Selection via the Elastic Net,” *Journal of the Royal Statistical Society*, Series B, 67, 301–320. [1053]