OJMO

Open Journal of Mathematical Optimization

Peng Zheng, Karthikeyan Natesan Ramamurthy & Aleksandr Y. Aravkin

Estimating Shape Parameters of Piecewise Linear-Quadratic Problems

Volume 2 (2021), article no. 8 (18 pages)

https://doi.org/10.5802/ojmo.10

Article submitted on December 30, 2020, revised on August 31, 2021,
accepted on August 30, 2021.

Open Journal of Mathematical Optimization is a member of the
Centre Mersenne for Open Scientific Publishing
http://www.centre-mersenne.org/

CENTRE MERSENNE

Estimating Shape Parameters of Piecewise Linear-Quadratic Problems

Peng Zheng
Department of Health Metrics Sciences, University of Washington
Seattle, WA, USA
zhengp@uw.edu

Karthikeyan Natesan Ramamurthy
IBM Thomas J. Watson Research Center
Yorktown Heights, NY, USA
knatesa@us.ibm.com

Aleksandr Y. Aravkin
Department of Applied Mathematics, University of Washington
Seattle, WA, USA
saravkin@uw.edu

Abstract

Piecewise Linear-Quadratic (PLQ) penalties are widely used to develop models in statistical inference, signal processing, and machine learning. Common examples of PLQ penalties include least squares, Huber, Vapnik, 1-norm, and their asymmetric generalizations. Properties of these estimators depend on the choice of penalty and its shape parameters, such as degree of asymmetry for the quantile loss, and transition point between linear and quadratic pieces for the Huber function.

In this paper, we develop a statistical framework that can help the modeler to automatically tune shape parameters once the shape of the penalty has been chosen. The choice of the parameter is informed by the basic notion that each PLQ penalty should correspond to a true statistical density. The normalization constant inherent in this requirement helps to inform the optimization over shape parameters, giving a joint optimization problem over these as well as primary parameters of interest. A second contribution is to consider optimization methods for these joint problems. We show that basic first-order methods can be immediately brought to bear, and design specialized extensions of interior point (IP) methods for PLQ problems that can quickly and efficiently solve the joint problem. Synthetic problems and larger-scale practical examples illustrate the utility of the approach. Code for the new IP method is implemented using the `Julia` language (`https://github.com/UW-AMO/shape-parameters/tree/ojmo`).

Digital Object Identifier 10.5802/ojmo.10

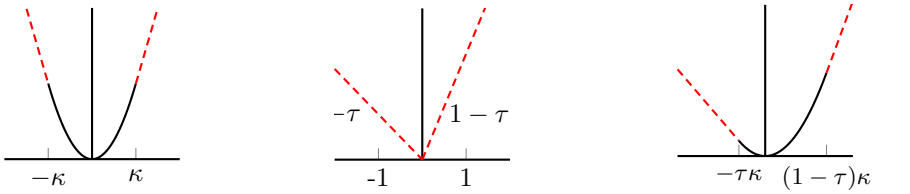
Acknowledgments The authors acknowledge the Washington Research Foundation Data Science Professorship.

1 Introduction

Piecewise Linear-Quadratic (PLQ) functions and their superclass, Quadratic Support functions [1] form a rich class of penalties that are used for a range of applications. The choice of penalty plays a key role in the properties of the final estimate. Using the Huber penalty (Figure 1 (a)) rather than a quadratic to measure data misfit is a typical approach to make an estimate robust to outliers [17]. The 1-norm applied directly to parameters makes the final solution sparse, and is used in Lasso regression [29] and compressed sensing [15]. The asymmetric quantile loss (see Figure 1 (b)) is used for many regression applications [20].

The correspondence of the *shape* of a PLQ function to its role in estimation is well-understood. For example, it is the linear tails of the Huber penalty that limit the effects of large residuals on the final estimate when used as a misfit, and it is the nonsmooth behavior at the origin of the 1-norm that promotes sparse solutions when used as a regularizer. Likewise, the asymmetry of the quantile loss that make it useful for financial applications, where loss and gain are treated asymmetrically.

Here, we consider the choice of parameters that fully specify this shape. At what value should the Huber transition between quadratic and linear? How asymmetric do we need the quantile loss to be? We focus on questions that can be answered once we have model estimates and corresponding residuals in hand, but seem as difficult a priori as any other parameter selection problem.



(a) Huber (κ) **(b)** quantile (τ) **(c)** quantile Huber (τ, κ)

Figure 1 Huber and quantile families parametrized by κ and τ, with the quantile Huber parameterized by both. Linear pieces are shown using red dash; curvilinear pieces are shown using black solid.

Unknown parameters in regression and inverse problems are usually estimated using cross-validation or grid search. Standard methods require multiple solutions of any given learning problem, where a held-out dataset is used to evaluate each configuration [5]. More recently, Bayesian optimization [9, 18, 19, 28] and random search [10, 22] have come to the forefront as two popular techniques used to obtain meta-parameters in a very wide range of contexts. While these techniques can be used for the problems we consider, as well as a much more general problems, their implementation is more expensive than solving a single problem instance (cross-validation, random search and Bayesian optimization require many instance evaluations).

Here we take a different tack, and build on the statistical perspective developed in [1] to develop a simple modification of the PLQ estimation problem to infer shape parameters simultaneously with variables of interest. The most relevant works related to this paper focus on the relation between the quantile penalty and the asymmetric Laplace distribution(ALD) [8, 30, 33]. In particular, [8] jointly estimate the model and the shape parameters for quantile penalty, and [30] infer the joint posterior distribution of these parameters.

To explain the general idea, we look at two simple examples.

1.1 Classic variance estimation in linear regression

As a warm-up, we consider the classic problem of estimating regression variables jointly with noise variance parameter in a linear Gaussian model. Take the basic statistical model

$$y = Ax + \epsilon, \quad \epsilon \sim N(0, \sigma^2 I), \quad i = 1, \dots, n, \tag{1}$$

we can find the maximum likelihood estimator for (x, σ^2) by maximizing

$$p(x, \sigma^2|y) \propto p(y|x, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp((y_i - \langle a_i, x\rangle)^2/2\sigma^2)$$

or equivalently by minimizing the negative log of this expression:

$$\min_{x,\sigma^2} \frac{1}{2\sigma^2}\|y - Ax\|^2 + \frac{n}{2}\log(\sigma^2). \tag{2}$$

In this simple case, the problem separates, and regardless of σ^2 we have

$$\widehat{x} = (A^T A)^{-1} A^T y.$$

Minimizing (2) with respect to σ^2, we also get

$$\widehat{\sigma}^2 = \|y - A\widehat{x}\|^2/n,$$

the average of empirical residuals squared. The main point to here is that the likelihood arising from the statistical model (1) enables us to estimate σ^2. The idea becomes much more interesting when parameters are coupled. For example, in pharmacokinetic modeling several different datasets with unknown scalar variance parameters may inform a shared parameter vector x [7]. Another example where the shape parameter affects the estimate of x is presented in the next section.

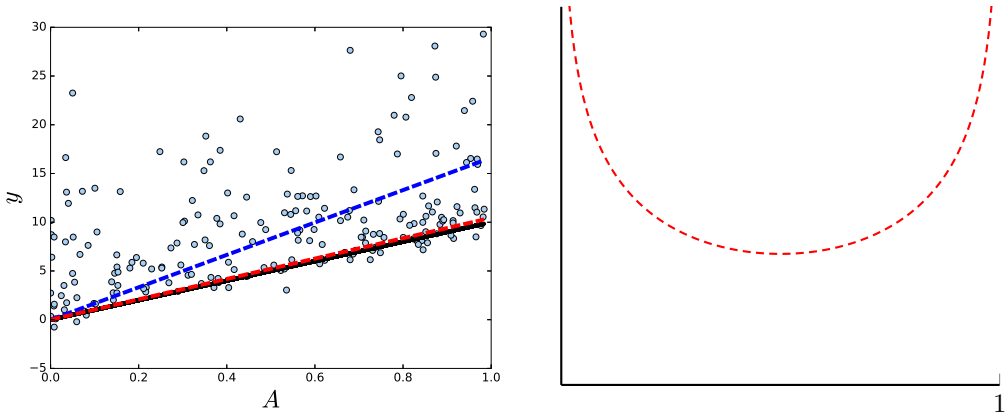


Figure 2 Left panel: regression from data with asymmetric errors. Data are shown in black; true model in black; linear regression estimate in black dash, and the new auto-tuned quantile estimate in red dash. Right: graph of $\log(1/\tau + 1/(1-\tau))$ in (4). This term, derived in Section 3, acts as a barrier pushing the quantile estimator into $(0, 1)$.

1.2 Simultaneous quantile estimation and regression.

Consider Figure 2, where linear observations have been contaminated with asymmetric errors, i.e. they are more likely to be positive than negative. The data generating mechanism is shown in black. The linear regression model for the data $\{y_i, a_i\}$ is equivalent to the least square problem,

$$\min_x \frac{1}{2}\|Ax - y\|^2$$

where x contains slope and intercept. As expected, classic regression fails to capture the true mechanism because the errors are asymmetric, see black dash in Figure 2.

In this case, we can actually recover the "true" line by assuming the errors arise from an asymmetric generalization of the Laplace distribution:

$$y_i = \langle a_i, x\rangle + \epsilon_i, \quad \epsilon_i \sim \exp(-\rho_\tau(\,\cdot\,)), \tag{3}$$

where ρ_τ is the penalty shown in Figure 1(b), with $\tau \in [0, 1]$ controlling the relative slopes to the left and right of the origin. Solving for a *joint* maximum likelihood estimator for (x, τ), we get the fit in red dash in Figure 2. The estimator is derived by considering the density corresponding to (3), and has the simple expression

$$\min_{x,\tau\in[0,1]} \rho_\tau(Ax - y) + m\log\left(\frac{1}{\tau} + \frac{1}{1-\tau}\right). \tag{4}$$

where the last term arises from considering the normalization constant required to make $\exp(-\rho_\tau(\,\cdot\,))$ a true statistical density, and is derived in Section 3, where the general approach is developed. In this simple example, the right solution is found by solving the single problem (4) once, rather than considering multiple problem instances as we expect from classic approaches. The optimization problem itself is nonsmooth and nonconvex, but as we will see straightforward. The main thrust of the paper is to get a systematic way of formulating estimators such as (4), understand their properties, and consider different algorithms for solving these problems.

1.3 Contributions and Roadmap

Building on the toolkit of PLQ penalties and corresponding densities developed in [1], we use the bridge between penalties and densities to develop an extended likelihood formulation over both shape parameters θ and regression variables x. The key idea is encoded in the *normalization constant*, that arises from the statistical interpretation and essentially acts as a balance against the data to ensure the final shape parameters still yield a statistically valid model. We consider properties of the extended likelihood formulations over (x, θ). In many cases, we show that standard techniques (including proximal alternating minimization and variable projection) can be applied. We also build on the interior point method of [1] to develop a new extended interior point (IP) approach tailored to these joint estimators. Code for the new IP method is made available in the `Julia` language[1].

[1] `https://github.com/UW-AMO/shape-parameters/tree/ojmo`

The paper proceeds as follows. In Section 2 we review PLQ functions, along with their conjugate representations, and examples. Conjugate representations are essential for solving PLQ models using interior point methods. In Section 3, we focus on the statistical interpretation, and derive the normalization constant needed to create joint estimators such as (4). In Section 4, we consider optimization methods that can be applied to the new class of estimators, and derive a new customized interior point method for the class. In Section 5 we provide synthetic verifications that show we can recover shape parameters and regression variables in simple examples. We empirically show that the new joint estimation approach is consistent, and performs as well as simple least squares estimation when errors are Gaussian. Section 6 contains some applications of these ideas to real datasets, particularly focusing on self-tuning penalties for robust PCA.

2 PLQ Functions and Their Conjugate Representations

Piecewise linear-quadratic (PLQ) functions [27, Definition 10.20] are convex functions whose domain can be represented as the union of finitely many polyhedral sets, relative to each of which the function can be written as a convex quadratic. These functions have a convenient representation using their convex conjugates.

▶ **Definition 1** (Convex conjugate). *The* convex conjugate *of a function f is given by*

$$f^*(v) = \sup_u u^\mathsf{T} v - f(u).$$

A PLQ function is defined as the conjugate to a quadratic over a polyhedral set [27, Example 11.18]. Examples of common penalties used in statistical modeling, machine learning, and inverse problems can all be written using simple conjugates, as shown in Table 1. Quadratic support (QS) functions [1] generalize PLQ functions by removing the polyhedral set restriction [1]. Here we focus on PLQ examples, since they are computationally tractable.

Table 1 Common PLQ Functions and their Conjugate Representations

Name	Conjugate Representation	Figure
Huber	$h_\kappa(x) = \sup_{u\in[-\kappa,\kappa]} \left\{ux - \frac{1}{2}u^2\right\}$	Fig. 1 (a)
Quantile	$q_\tau(x) = \sup_{u\in[-\tau,(1-\tau)]} \{ux\}$	Fig. 1 (b)
Quantile Huber	$h_{\tau,\kappa}(x) = \sup_{u\in[-\kappa\tau,\kappa(1-\tau)]} \left\{ux - \frac{1}{2}u^2\right\}$	Fig. 1 (c)
Least squares	$\frac{1}{2}x^2 = \sup_u \left\{ux - \frac{1}{2}u^2\right\}$	Fig. 3 (a)
Hinge	$h_\epsilon(x) = \sup_{u\in[-\tau,(1-\tau)]} \{u(x-\epsilon)\}$	Fig. 3 (b)
Vapnik	$\rho_\epsilon(x) = \sup_{u\in[0,1]^2} \left\{\left\langle \begin{bmatrix}1\\-1\end{bmatrix} x - \begin{bmatrix}\epsilon\\\epsilon\end{bmatrix}, u \right\rangle\right\}$	Fig. 3 (c)
Smooth insensitive loss	$\rho^h_\epsilon(x) = \sup_{u\in[0,1]^2} \left\{\left\langle \begin{bmatrix}1\\-1\end{bmatrix} x - \begin{bmatrix}\epsilon\\\epsilon\end{bmatrix}, u \right\rangle - \frac{1}{2}u^T u\right\}$	Fig. 3 (d)
Elastic net	$\rho(x) = \sup_{u\in[0,1]\times\mathbb{R}} \left\{\left\langle \begin{bmatrix}1\\1\end{bmatrix} x, u \right\rangle - \frac{1}{2}u^T \begin{bmatrix}0&0\\0&1\end{bmatrix} u\right\}$	Fig. 3 (e)

We can explicitly define PLQ functions using their conjugate representation.

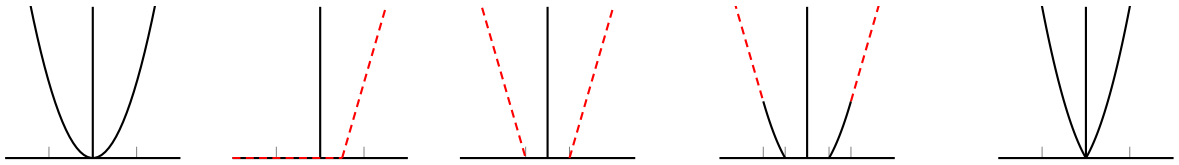
▶ **Definition 2** (PLQ Functions and penalties). *A piecewise linear-quadratic (PLQ) function is given by*

$$\rho(r) = \sup_{u\in U} \left\{u^\mathsf{T}(Br - \bar{b}) - \frac{1}{2}u^\mathsf{T} M u : C^\mathsf{T} u \le \bar{c}\right\}, \tag{5}$$

where $M \succeq 0$, B is an injective linear map, and $U := \{u : C^\mathsf{T} u \le \bar{c}\}$ is a polyhedral set. When $0 \in U$, it follows immediately that ρ must be non-negative; and in this case we call it a penalty.

PLQ functions are closed under sums and affine compositions [1], and have a representation calculus that makes it easy to conjugate for a PLQ problem that combines several terms, such as measurement models and regularizers.

A statistical interpretation of PLQs requires the notion of *coercivity*.



(a) quadratic **(b)** hinge, $\epsilon = 0.5$ **(c)** Vapnik, $\epsilon = 0.5$ **(d)** Smooth insensitive loss **(e)** elastic net ($\alpha = 0.5$)

Figure 3 Five PLQ penalties frequently used in learning and inverse problems. Linear portions are shown in red dash and curvilinear pieces are shown in black solid. Huber and quantile losses are shown in Fig. 1.

▶ **Definition 3** (Coercivity). *A function $f(x)$ is coercive if*

$$\liminf_{\|x\|\to\infty} f(x) = \infty.$$

Coercivity of PLQs can be expressed in terms of their conjugate representations, and this topic is addressed by [1, Theorem 10]. In the next section, we use coercive PLQs to define corresponding densities, which enable the joint likelihood approach developed in this paper.

3 Statistical Model and Properties of Joint Objective

To formulate a learning problem over spaces of penalties, we first review the relationship between penalties and corresponding residual distributions. We then use this relationship to develop a joint formulation for model and shape parameter inference, and characterize properties of the resulting objective function.

The relationship between penalties and associated densities can be made precise. Every coercive PLQ penalty can be viewed as the negative logarithm of an integrable density. Specifically, given a coercive penalty $\rho(r;\theta)$ whose shape is parametrized by θ, we define an associated density

$$p(r;\theta) = \frac{1}{n_c(\theta)} \exp[-\rho(r;\theta)], \quad \text{where} \quad n_c(\theta) \quad := \int_{\mathbb{R}} \exp[-\rho(r;\theta)]\, \mathrm{d}r < \infty. \tag{6}$$

The term $n_c(\theta)$ is a normalization constant that ensures that $p(r;\theta)$ in (6) is a true statistical density, i.e. it integrates to 1. These observations yield a simple recipe for extending a negative log likelihood in x (based on optimizing a PLQ penalty) to inform shape parameters θ. Specifically, given an optimization problem of the form

$$\min_x \sum_{i=1}^m \rho(y_i - \langle a_i, x\rangle\,;\theta)$$

we consider the extended problem

$$\min_{x,\theta\in\mathcal{D}} \sum_{i=1}^m \rho(y_i - \langle a_i, x\rangle\,;\theta) + m\log[n_c(\theta)], \tag{7}$$

where θ may be restricted to a domain $\mathcal{D}$. In the least squares case, this idea reduces to the classic variance estimation example in Section 1.1. The quantile penalty example in Section 1.2 recovers the formulation studied by [8]. The general approach is a new way to estimate shape parameters of error distributions corresponding to PLQ penalties.

The quantile regression problem in Section 1.2 yields a closed form solution for $n_c(\tau)$. We have $\mathcal{D} = [0, 1]$ and

$$\rho_\tau(x) = \begin{cases} (1-\tau)x & x \geq 0 \\ -\tau x & x \leq 0 \end{cases}, \quad p(r;\tau) = \begin{cases} \exp(-(1-\tau)x) & x \geq 0 \\ \exp(-\tau x) & x \leq 0 \end{cases}, \quad n_c(\tau) = \frac{1}{1-\tau} + \frac{1}{\tau}.$$

The $\log(n_c)$ term acts as a barrier (see Figure 2), pushing τ away from the boundary points 0 and 1 into the interior $(0, 1)$, and favoring $\tau = 0.5$.

Considering the problem from an algorithm design perspective, the log of the normalization constant for the quantile regression problem is smooth and strongly convex function of the shape parameter τ, but its gradient is not Lipschitz continuous. In the remainder of this section, we characterize theoretical properties of the general objective (7).

▷ Assumption 4. To ensure the validity of the statistical viewpoint, we require ρ to satisfy:

1. $\rho(r;\theta) \geq 0$ is a PLQ *penalty* for every $\theta \in \mathcal{D}$, so we have *non-negativity.*
2. ρ is *coercive* for any $\theta \in \mathcal{D}$, so we have *integrability.*
3. For any $\theta_0 \in \mathcal{D}$, $\rho(r;\theta)$ is C^2 around θ_0 for almost every $r \in \mathbb{R}$ (*smoothness in* θ)

Under these assumptions, we can obtain formulas for the first and second derivatives of $n_c(\theta)$.

▶ **Theorem 5** (smoothness of $n_c(\theta)$)**.** *For $n_c(\theta)$in (6), suppose Assumption 4 holds and for $\theta_0 \in \mathcal{D}$, there exist functions $g_k(r)$, $k = 1, 2$, such that,*

1. *for any unit vector v, $|\langle \nabla_\theta \exp[-\rho(r;\theta)], v\rangle| \leq g_1(r)$ for any θ around θ_0,*
2. *for any unit vector v, $|\langle \nabla^2_\theta \exp[-\rho(r;\theta)]v, v\rangle| \leq g_2(r)$ for any θ around θ_0,*
3. *$\int_{\mathbb{R}} g_k(r)\,\mathrm{d}r < \infty$, $k = 1, 2$.*

Then $n_c(\theta)$ is C^2 continuous around θ_0 and,

$$\nabla n_c(\theta_0) = \int_{\mathbb{R}} \nabla_\theta \exp[-\rho(r;\theta_0)]\,\mathrm{d}r, \qquad \nabla^2 n_c(\theta_0) \quad = \int_{\mathbb{R}} \nabla^2_\theta \exp[-\rho(r;\theta_0)]\,\mathrm{d}r. \tag{8}$$

The proof is elementary, and is included below for completeness.

Proof. From Assumption 4, we know that for any $\theta_0 \in \mathcal{D}$, $\nabla_\theta \exp[\rho(r;\theta_0)]$ and $\nabla^2_\theta \exp[\rho(r;\theta_0)]$ exist for almost every $r \in \mathbb{R}$. For any h such that $\|h\|$ is small enough to make $\theta_0 + h$ stay in the neighborhood of θ_0. Applying the mean value theorem, we have,

$$n_c(\theta_0 + h) - n_c(\theta_0) = \int_{\mathbb{R}} \exp[-\rho(r;\theta_0 + h)] - \exp[-\rho(r;\theta_0)]\,\mathrm{d}r = \int_{\mathbb{R}} \left\langle \nabla_\theta \exp[-\rho(r;\bar{\theta})], h \right\rangle \mathrm{d}r$$
$$\implies \frac{n_c(\theta_0 + h) - n_c(\theta_0)}{\|h\|} = \int_{\mathbb{R}} \left\langle \nabla_\theta \exp[-\rho(r;\bar{\theta})], \frac{h}{\|h\|} \right\rangle \mathrm{d}r$$

where $\bar{\theta}$ lie in segment with the end points θ_0 and $\theta_0 + h$. The first and third assumptions allow us to apply the dominated convergence theorem to get

$$\lim_{h\to 0} \frac{n_c(\theta_0 + h) - n_c(\theta_0)}{\|h\|} = \lim_{h\to 0} \int_{\mathbb{R}} \left\langle \nabla_\theta \exp[-\rho(r;\bar{\theta})], \frac{h}{\|h\|} \right\rangle \mathrm{d}r = \int_{\mathbb{R}} \lim_{h\to 0} \left\langle \nabla_\theta \exp[-\rho(r;\bar{\theta})], \frac{h}{\|h\|} \right\rangle \mathrm{d}r$$
$$= \int_{\mathbb{R}} \langle \nabla_\theta \exp[-\rho(r;\theta_0)], v\rangle\,\mathrm{d}r = \left\langle \int_{\mathbb{R}} \nabla_\theta \exp[-\rho(r;\theta_0)]\,\mathrm{d}r, v \right\rangle$$

where we set $h = \alpha v$ and let $\alpha \to 0^+$ and keep v fix as an unit vector. From the definition of the gradient we know that,

$$\nabla n_c(\theta_0) = \int_{\mathbb{R}} \nabla_\theta \exp[-\rho(r;\theta_0)]\,\mathrm{d}r.$$

Following the same steps we can show $\nabla^2 n_c(\theta_0)$ exists and satisfies

$$\nabla^2 n_c(\theta_0) = \int_{\mathbb{R}} \nabla^2_\theta \exp[-\rho(r;\theta_0)]\,\mathrm{d}r. \qquad \blacktriangleleft$$

The derivative formulas (8) are used for first- and second-order methods to infer x and θ. The parametrization conditions in θ are satisfied by all piecewise linear-quadratic (PLQ) penalties. The theorem can also be applied to densities that are not log-concave. For instance, the Student's t density and associated penalty satisfy all assumptions of Theorem 5.

In the quantile case (4), the term $\log[n_c(\theta)]$ is convex. We characterize sufficient conditions for convexity of $\log[n_c(\theta)]$ for a general class of penalties ρ. The results can be derived using [12, Chapter 3.5].

▶ **Theorem 6** (convexity of $\log[n_c(\theta)]$)**.** *Let $n_c(\theta)$ be defined as in Theorem 5, and suppose Assumption 4 holds. We have the following results:*

1. *If $\rho(r;\theta)$ is jointly convex in r and θ, then $\log[n_c(\theta)]$ is a concave function of θ.*
2. *If $\rho(r;\theta)$ is concave with respect to θ for every r, then $\log[n_c(\theta)]$ is a convex function.*

Theorems 5 and 6 tell an interesting story. The log-normalization constant $\log[n_c(\theta)]$ is nearly always smooth, even when the loss ρ is nonsmooth in x. The inference problem (7) is *never guaranteed to be jointly convex* in (x, θ): if $\rho(x;\theta)$ is jointly convex, then $\log[n_c(\theta)]$ will be *concave*. This is intuitive, as we are attempting to learn both the model and error structure at the same time. Objective (7) is generally nonsmooth and nonconvex. In the next section, we show how to optimize it using first and second order methods.

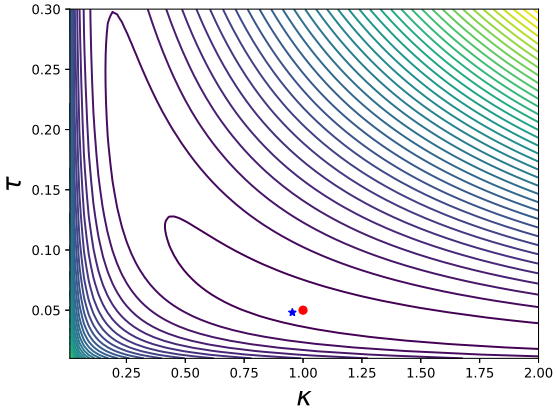


Figure 4 Level sets for value function $v(\theta)$ (9) for the quantile Huber model. The black star is the maximum likelihood estimator, while the red dot represents the true parameters in the simulation.

3.1 Level sets of the objective function and maximum likelihood estimate

Although (7) is non-convex, in many cases we can still find the global minimum in θ. To illustrate, we generate the samples ϵ_i from the distribution defined by the quantile Huber function with $\kappa = 1$ and $\tau = 0.05$, select a set of weights x, generate data $y_i = \langle a_i, x\rangle + \epsilon_i$, and plot the so called *value function* for (7):

$$v(\theta) = \min_x \sum_{i=1}^{m} \rho(y_i - \langle a_i, x\rangle ; \theta) + m \log[n_c(\theta)]. \tag{9}$$

Evaluating $v(\theta)$ requires solving a smooth convex problem in x, and thus problem (7) reduces to minimizing (9) in θ. Results for the simple 2D case are shown in Figure 4. $v(\theta)$ is non-convex (note the level sets), but there is a unique minimum that is close to the true parameters, and was found in every run by a local search. In the next section, we design methods that are far more efficient than optimizing $v(\theta)$.

4 Optimization Methods for Joint PLQ and Shape Parameter Estimation.

In this section, we discuss algorithms for (7), and develop a new interior point method building on the PLQ optimization strategy of [1]. When ρ is smooth in x and θ, we show how to apply Proximal Alternating Linearized Minimization (PALM) [11] and Proximal Alternating Minimization (PAM) [6]. We also discuss variable projection (VP) type algorithms [3, 4]. All three algorithms can be applied to the joint estimation problem, as long as we carefully consider the problem structure. PAM has a disadvantage compared to the other two, since it requires a fast solver for a regularized PLQ problem. Such a solver was developed in [1], but in this case we might as well use the new IP method developed in Section 4.2. In contrast, PALM can be readily implemented, as long as we have smoothness in the term that couples x and θ; we use PALM for the large-scale experiments in Section 6. One catch is that the log normalization constant $\log[n_c(\theta)]$ must be treated carefully, as its gradient does not have a global Lipschitz constant. The VP variant we consider here is also easy to implement, since it focuses on projecting out θ rather than x.

In Section 4.2 we extend interior point methods developed in [1] to problem (7). These methods can simultaneously fit PLQ problems (without e.g. smoothness requirements) and estimate their shape parameters. The reader will recall that the quantile regression example in Section 1.2 uses a nonsmooth PLQ penalty ρ. The extended IP approach can solve nonconvex problems with primal, conjugate, dual, and shape parameters together by directly working with the associated system of optimality conditions. We have released the implementation of the method in the `Julia` language.

4.1 PALM, PAM, and Variable Projection

Several algorithms from existing literature can be brought to bear on problem (7) when it is of the form

$$\min_{x,\theta} H(x, \theta) + r_1(x) + r_2(\theta), \tag{10}$$

under some mild assumptions on H. Here we briefly review PALM [11], and PAM [6] algorithms, as well as similar algorithms that partially minimize the objective in θ, known as *partial minimization* or *variable projection* [3, 4].

The PALM and PAM algorithms [11] can be used to minimize (10) when H is C^1, with globally Lipschitz partial gradients, while the functions r_1 and r_2 are proper lower semicontinuous; in particular they are not required to be convex, finite-valued, or smooth. The term $r_1(x)$ is useful if we need simple regularization and constraints on x, such as sparsity or non-negativity. Even though $\log[n_c(\theta)]$ is smooth (see Theorem 5), it must be relegated to $r_2(\theta)$, since otherwise it violates the Lipschitz assumptions on the partial gradients of H. Therefore, to apply PALM or PAM to (7), we take

$$H(x,\theta) = \sum_{i=1}^{m} \rho(y_i - \langle a_i, x\rangle\,; \theta), \qquad r_2(\theta) = \delta_{\mathcal{D}}(\theta) + m\log[n_c(\theta)]. \tag{11}$$

Here $\delta_{\mathcal{D}}$ is the indicator function for the set $\mathcal{D}$,

$$\delta_{\mathcal{D}}(\theta) = \begin{cases} 0 & \text{if } \theta \in \mathcal{D}, \\ \infty & \text{if } \theta \notin \mathcal{D}. \end{cases}$$

The PALM algorithm is detailed in Algorithm 1. The steps c_k and d_k are obtained from Lipschitz constants of the (partial) gradients of H. The PAM algorithm is given in Algorithm 2, with the same step sizes to facilitate an easier comparison.

PAM essentially requires the ability to partially minimize a quadratically regularized problem H with respect to both x and θ. In many cases, one of these variables may be easier to solve for than the other; in Example 3.1, θ has dimension 2. Variable projection algorithms [3, 4] exploit the ability to fully minimize in one variable while applying an iterative approach in the other. Algorithm 3 shows an example where we have chosen to optimize out θ for every iteration in x.

Algorithm 1 PALM for (11)

Require: A, y

Initialize: x^0, θ^0

while not converge **do**

$x^{k+1} \leftarrow \operatorname{prox}_{\frac{1}{c_k} r_1}\left(x^k - \frac{1}{c_k}\nabla_x H(x^k, \theta^k)\right)$

$\theta^{k+1} \leftarrow \operatorname{prox}_{\frac{1}{d_k} r_2}\left(\theta^k - \frac{1}{d_k}\nabla_\theta H(x^{k+1}, \theta^k)\right)$

return x^k and θ^k

Algorithm 2 PAM for (11)

Require: A, y

Initialize: x^0, θ^0

while not converge **do**

$x^{k+1} \leftarrow \min_x H(x, \theta^k) + r_1(x) + \frac{1}{c_k}\|x - x^k\|^2$

$\theta^{k+1} \leftarrow \min_\theta H(x^{k+1}, \theta) + r_2(\theta) + \frac{1}{d_k}\|\theta - \theta^k\|^2$

return x^k and θ^k

Algorithm 3 VP for (11), partially minimizing w.r.t. θ

Require: A, y

Initialize: x^0, θ^0

while not converge **do**

$x^{k+1} \leftarrow \operatorname{prox}_{\frac{1}{c_k} r_1}\left(x^k - \frac{1}{c_k}\nabla_x H(x^k, \theta^k)\right)$

$\theta^{k+1} \leftarrow \min_\theta H(x^{k+1}, \theta) + r_2(\theta)$

return x^k and θ^k

Detail. The prox operator of $\log[n_c(\theta)]$ is not available in closed form for any examples of interest. Instead, it can be efficiently computed using the results of Theorem 5:

$$\operatorname{prox}_{\frac{1}{d_k}r_2}(\phi) = \arg\min_{\theta\in\mathcal{D}} \frac{1}{2d_k}\|\theta-\phi\|^2 + \log[n_c(\theta)]. \tag{12}$$

In all examples of interest, θ is low dimensional; and we compute (12) using Newton's method or an interior point method. This requires $\nabla \log[n_c(\theta)]$ and $\nabla^2 \log[n_c(\theta)]$, which are calculated numerically using formulas (8).

The PALM algorithm works well for large-scale shape inference problems with smooth ρ. We use it for the self-tuning RPCA experiments in Section 6.

4.2 Interior Point method for nonsmoothly coupled nonconvex joint QS inference

In this section, we use conjugate representations of PLQ penalties to develop an interior point method for the extended inference problem (7):

$$\min_{x,\theta\in\mathcal{D}} \sum_{i=1}^{m} \rho(y_i - \langle a_i, x\rangle\,;\theta) + \log[n_c(\theta)].$$

The approach converges superlinearly in practice, but each iteration requires solving a linear system. We solve these systems directly, and scaling issues are explored in Table 3. In practice, large-scale linear systems are solved iteratively, often using pre-conditioners [25]; we leave these developments for future work.

To optimize PLQ penalties parametrized by θ, we allow $\bar{b}$ and $\bar{c}$ in (5) to be affine functions of θ, and assume $\mathcal{D}$ is also polyhedral:

$$\bar{b} = G^\mathsf{T}\theta + b, \quad \bar{c} = H^\mathsf{T}\theta + c, \quad \mathcal{D} = \{\theta : S^\mathsf{T}\theta \le s\}.$$

We then solve a saddle point for primal variables x, conjugate variables u, and shape parameters θ:

$$\min_{x,S^\mathsf{T}\theta\le s} \sup_{u} \left\{ u^\mathsf{T}[B(Ax-y) - G^\mathsf{T}\theta - b] - \frac{1}{2}u^\mathsf{T}Mu : C^\mathsf{T}u \le H^\mathsf{T}\theta + c \right\} + m\log[n_c(\theta)] \tag{13}$$

For example, the self-tuning quantile penalty (4) is written as

$$\min_{x,\tau\in\mathcal{C}_1} \sup_{(u,\tau)\in\mathcal{C}_2} \left\{ u^\mathsf{T}(Ax-b) + m\log\left(\frac{1}{\tau} + \frac{1}{1-\tau}\right)\right\}.$$

$$\mathcal{C}_1 := \left\{ \tau : \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tau \le \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\}, \quad \mathcal{C}_2 = \left\{ (u,\tau) : \begin{bmatrix} 1 \\ -1 \end{bmatrix} u \le -\begin{bmatrix} 1 \\ 1 \end{bmatrix} \tau + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\}.$$

Algorithm 4 Interior point method for QS with θ estimation

Require: A, y, B, b, C, c, G, H, S, s
1: **Initialize**: z^0, $k \leftarrow 0$, $\mu = 1$
2: **while** not converged **do**
3: $\quad p \leftarrow \nabla F_\mu(z^k)^{-1} F_\mu(z^k)$
4: $\quad \alpha \leftarrow \text{LineSearch}(z^k, p)$, using $\|F_\mu(\,\cdot\,)\|$
5: $\quad z^{k+1} \leftarrow z^k - \alpha p$
6: $\quad \mu \leftarrow 0.1\cdot$ (Average complementarity conditions)
return z^{k+1}

Interior point (IP) methods apply damped Newton to a relaxation of the optimality conditions of (13), see [21, 24, 32]. Let $F(z)$ denote the optimality conditions for problem (13), with $z = (x, u, \theta, \lambda)$, where λ are dual variables for inequality constraints $C^\mathsf{T}u \le H^\mathsf{T}\theta + c$ and $S^\mathsf{T}\theta \le s$, and let $F_\mu(z)$ denote the relaxed system obtained by using a logarithmic barrier with parameter μ. The IP method is summarized in Algorithm 4.

We introduce a log-barrier for the conjugate variables u and shape parameters θ as follows.

$$\delta_{\mathcal{D}}(\theta) \approx -\mu \mathbb{1}^\mathsf{T}\log(s - S^\mathsf{T}\theta),$$

$$\delta_{\{(u,\theta)|C^\mathsf{T}u\le H^\mathsf{T}\theta+c\}}(u,\theta) \approx -\mu\mathbb{1}^\mathsf{T}\log(c + H^\mathsf{T}\theta - C^\mathsf{T}u).$$

As $\mu \downarrow 0$, the barriers approach true indicator functions for the $\mathcal{D}$ and U. The parameter μ is decreased to a specified tolerance as the optimization proceeds. For fixed μ, the approximate subproblem with fixed μ can itself be written as a saddle point system:

$$\min_{x,\theta} \sup_{u} \left\{ u^\mathsf{T}[B(Ax-y)-G^\mathsf{T}\theta-b]-\frac{1}{2}u^\mathsf{T}Mu+\mu\mathbb{1}^\mathsf{T}\log(c+H^\mathsf{T}\theta-C^\mathsf{T}u) \right\}+m\log[n_c(\theta)]-\mu\mathbb{1}^\mathsf{T}\log(s-S^\mathsf{T}\theta) \quad (14)$$

We introduce dual variables q and slack variables d:

$$d := \begin{bmatrix} c \\ s \end{bmatrix} - \begin{bmatrix} C^\mathsf{T} & -H^\mathsf{T} \\ 0 & S^\mathsf{T} \end{bmatrix} \begin{bmatrix} u \\ \theta \end{bmatrix}, \quad D := \mathrm{Diag}(d), \quad q := \mu D^{-1}\mathbb{1}, \qquad Q := \mathrm{Diag}(q), \quad z := [q, u, x, \theta]^\mathsf{T}$$

where the Diag operator acting on a vector v returns a diagonal matrix with diagonal v. We can then form the KKT system

$$F_\mu(z) = \begin{bmatrix} Dq - \mu\mathbb{1} \\ B(Ax-y) - G^\mathsf{T}\theta - b - Mu + \begin{bmatrix} -C & 0 \end{bmatrix} q \\ A^\mathsf{T}B^\mathsf{T}u \\ -Gu + m\nabla\log[n_c(\theta)] + \begin{bmatrix} H & S \end{bmatrix} q \end{bmatrix} \quad (15)$$

The Jacobian matrix $F_\mu^{(1)}$ of the system is given by

$$\nabla F_\mu(z) = \left[\begin{array}{c|c|c|c} D & Q\begin{bmatrix} -C^\mathsf{T} \\ 0 \end{bmatrix} & & Q\begin{bmatrix} H^\mathsf{T} \\ -S^\mathsf{T} \end{bmatrix} \\ \hline \begin{bmatrix} -C & 0 \end{bmatrix} & -M & BA & -G^\mathsf{T} \\ \hline & A^\mathsf{T}B^\mathsf{T} & & \\ \hline \begin{bmatrix} H & S \end{bmatrix} & -G & & m\nabla^2\log(n_c) \end{array}\right] \quad (16)$$

Algorithm 4 is a damped Newton iteration to find the stationary point of F_μ. At each iteration, μ is taken to be a fraction of the current average complementarity conditions, just as in the implementation used in [1, 2].

We can apply block Gaussian elimination to obtain conditions that make the algorithm implementable. We state the result as a simple theorem.

▶ **Theorem 7** (IP implementability). *Suppose that the PLQ penalty is nondegenerate, that is* $\mathrm{null}(C) \cap \mathrm{null}(M) = \{0\}$*, and that the linear model A is full-rank. Then the interior point iteration*

$$p = \nabla F_\mu(z^k)^{-1} F_\mu(z^k)$$

is implementable when a certain square symmetric system is invertible:

$$\begin{aligned} T_3 := {} & m\nabla^2\log(n_c) - HQH^\mathsf{T} + SQS^\mathsf{T} + (-G + HD^{-1}QC^\mathsf{T})T_1^{-1}(-G^\mathsf{T} + CD^{-1}QGH^\mathsf{T}) \\ & - (-G + HD^{-1}QC^\mathsf{T})T_1^{-1}BAT_2^{-1}A^\mathsf{T}B^\mathsf{T}T_1^{-1}(-G^\mathsf{T} + CD^{-1}QGH^\mathsf{T}) \end{aligned} \quad (17)$$

T_3 is a symmetric square matrix with dimension equal to the length of the parameter vector θ.

Proof. Implementing the IP iteration is equivalent to applying block-Gaussian elimination to the system (16). The set of operations is given by

$$\begin{aligned} R_2 &= R_2 + \begin{bmatrix} C & 0 \end{bmatrix} D^{-1}R_1 \\ R_4 &= R_4 - \begin{bmatrix} H & S \end{bmatrix} D^{-1}R_1 \\ R_3 &= R_3 + A^\mathsf{T}B^\mathsf{T}T_1^{-1}R_2 \\ R_4 &= R_4 + (-G + HD^{-1}QC^\mathsf{T})T_1^{-1}R_2 \\ R_4 &= R_4 - (-G + HD^{-1}QC^\mathsf{T})T_1^{-1}BAT_2^{-1}R3 \end{aligned}$$

where we define

$$T_1 := M + CD^{-1}QC^\mathsf{T}$$
$$T_2 := A^\mathsf{T}B^\mathsf{T}T_1^{-1}BA$$

The invertibility of D is enforced by Algorithm 4, which keeps slack variables d strictly positive. Invertibility of T_1 is guaranteed by the nondegeneracy hypothesis for the PLQ, see [1, Theorem 14]. Since B must be injective (see Definition 5), the full rank condition on A guarantees that T_2 is invertible. The row operations yield an upper triangular matrix of form

$$\nabla F_\mu(z) = \left[\begin{array}{c|c|c|c} D & Q\begin{bmatrix}-C^\mathsf{T}\\ 0\end{bmatrix} & & Q\begin{bmatrix}H^\mathsf{T}\\ -S^\mathsf{T}\end{bmatrix} \\ \hline & -T_1^\mathsf{T} & BA & -G^\mathsf{T} + CD^{-1}QGH^\mathsf{T} \\ \hline & & T_2 & A^\mathsf{T}B^\mathsf{T}T_1^{-1}(-G^\mathsf{T} + CD^{-1}QGH^\mathsf{T}) \\ \hline & & & T_3 \end{array}\right], \tag{18}$$

which is invertible if and only if T_3 in (17) is invertible, given the other hypotheses. ◀

The full rank condition on A may seem restrictive. However, in settings with rank deficient systems (e.g. in variable selection problems), the matrix A is naturally augmented to include the (PLQ) regularizer. For example, in Lasso, the 1-norm of x is part of the objective, and so the PLQ linear system is a stack the measurement matrix with two copies of the identity [1, Remark 5]. Thus, a rank-deficient A in the PLQ construction always points to an ill-posed statistical model.

In the next section, we present synthetic examples and that show the applicability of the approach for inferring simple shape parameters.

5 Synthetic Data Experiments

We illustrate the proposed approach using a linear regression example. We work in a simple regression setting, focusing on the quantile Huber family (Figure 1). Where appropriate, we make comparisons with least squares and least absolute deviation (LAD) formulations. Least squares estimates have a closed form solution, while we compute LAD estimates using `cvx.jl`.

In Section 5.1 we discuss the parametrization of the quantile Huber family. In Section 5.2 study consistency of the estimates for x in the Gaussian regime, as well as for errors generated from a particular quantile Huber distribution. In particular, we compare estimates as the number of observations increase with those obtained by for the least squares solution and LAD estimates. In Section 5.3, we consider a moderate number of observations, and compare the performance of the quantile Huber approach to those of least squares and LAD across a range of different quantile Huber parameters. In Section 5.4 we compare the performance of PALM to that of the interior point method for the quantile Huber problem. Finally, in Section 5.5 we compare the new approach to a grid search, a simple generic method applicable to the problem but unaware of the special problem structure.

5.1 Parametrization of the quantile Huber family

The quantile huber family is parametrized by τ, which controls slope of the penalty, and κ, which is the robustness threshold. We want to fit the regression model x as well as obtain the correct parameters τ and κ. When $\kappa > 0$ in quantile Huber, $\rho(x;\theta)$ is smooth, and we can use the PALM algorithm from Section 4.1. The quantile Huber penalty is PLQ, so we can also apply the proposed IP method from Section 4.2.

The primal form for the quantile Huber penalty is

$$\rho\left(r;\begin{bmatrix}\tau\\ \kappa\end{bmatrix}\right) = \begin{cases} -\tau\kappa r - \frac{(\tau\kappa)^2}{2}, & r < -\tau\kappa \\ \frac{1}{2}r^2, & r \in [-\tau\kappa, (1-\tau)\kappa] \\ (1-\tau)\kappa r - \frac{((1-\tau)\kappa)^2}{2}, & r > (1-\tau)\kappa \end{cases} = \begin{cases} -\theta_1 r - \frac{\theta_1^2}{2}, & r < -\theta_1 \\ \frac{1}{2}r^2, & r \in [-\theta_1, \theta_2] \\ \theta_2 r - \frac{\theta_2^2}{2}, & r > \theta_2 \end{cases} =: \rho\left(r;\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}\right) \tag{19}$$

We must choose a parametrization in terms of θ. One option is to take $\theta = [\tau, \kappa]^\mathsf{T}$. This parametrization could not be solved using the interior point approach, since the shape parameters would not be affine functions of θ.

It would also require a modification of the PALM algorithm, since the θ block as given would not have a global Lipschitz constant. An interesting insight here is that we can find an affine reparametrization of the shape parameters, which allows the problem to be solved using the interior point method we developed and satisfies the assumptions of the basic PALM algorithm. Specifically, we can take $\theta_1 = \tau\kappa, \theta_2 = \tau(1-\kappa)$. These new parameters must be non-negative.

The primal objective can be written as

$$\min_{x,\theta\geq 0} \sum_{i=1}^{m} \rho(y_i - \langle a_i, x\rangle ; \theta) + m \log[n_c(\theta)],$$

where $a_i \in \mathbb{R}^m$ are random Gaussian vectors, $x \in \mathbb{R}^n$ is the model parameter vector, and $y \in \mathbb{R}^m$ is the observed data vector contaminated by outliers. From Theorem 5, $n_c(\theta)$ is $\mathcal{C}^2$ smooth. From Theorem 6, the objective in θ is the sum of a concave term $\rho(Ax - y; \theta)$ and a convex term $m \log[n_c(\theta)]$. The joint problem in (x, θ) is nonconvex, but both first- and second-order methods from Section 4 can be applied.

5.2 Consistency

In this section, we test the performance of the estimation framework as the number of observations increases, for both Gaussian and quantile Huber errors. In both regimes, we want to estimate regression parameters x more accurately as the number of observations increases. For the Gaussian case, the idea is to check that performance is reasonable to ensure the framework is applicable in standard cases, i.e. in the absence of outliers or asymmetry. For quantile Huber, we want to know that we indeed recover all parameters as we see more observations. We consider $x \in \mathbb{R}^n$ for $n = 50$, and evaluate the performance of the method for $m \in \{1000, 1500, \ldots, 4500, 5000\}$. Errors in all experiments are generated from the $N(0, 1)$ distribution.

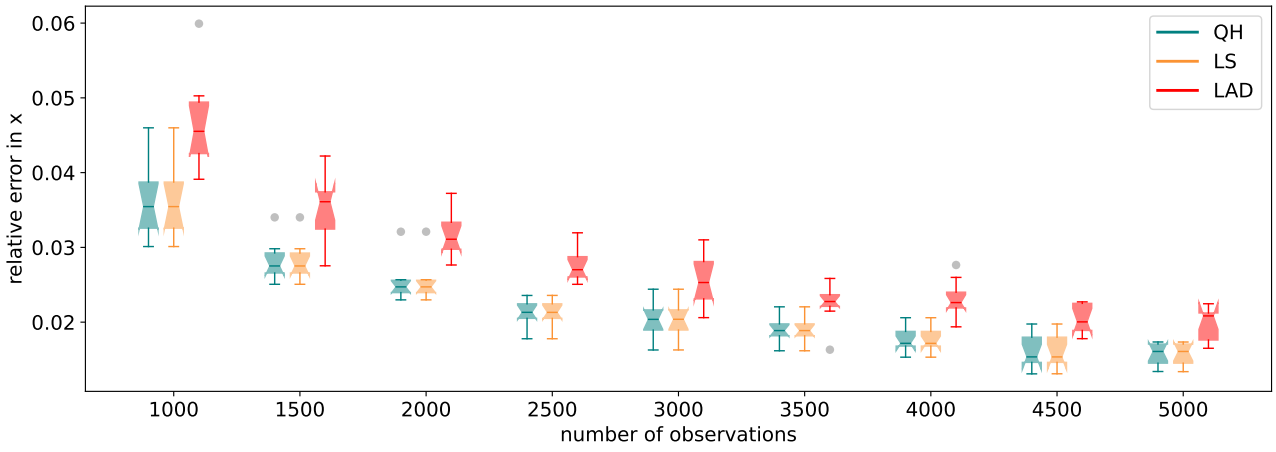


Figure 5 Relative error in recovery of regression parameters x for quantile huber (QH), least squares (LS) and least absolute deviation (LAD) approaches when generated measurement errors are $N(0, 1)$. Box plots summarize results across 10 realizations of each experiment. The quality of quantile Huber results is identical to that of the least squares solution, which is optimal in this nominal case. LAD solutions are comparable but always worse in terms of relative error.

We look at the relative error in inferred x across 10 realizations for each experiment, and plot this relative error as a function of the number of observations m in Figure 5. The quality of the quantile Huber solution is on par with the least squares solution, which is optimal in this case. The LAD solution is worse in comparison to the other approaches, but also improves with additional observations as expected.

In the nominal case, the inferred κ parameter in all cases pushes all residuals into the "inlier" region. To show this, we plot the minimum quadratic region, compared to the 95% confidence interval for the true residuals in Figure 6. The minimal quadratic region discovered across any problem realization is above 3, which means all of the residuals fall into the quadratic region of the objective function. This result also means values of τ are irrelevant, and uninformed by the estimated residuals. This is evident from model results; while τ estimates still center at 0.5, the estimates vary widely regardless of the available measurements, with a 95% interval from 0.2 to 0.7 across the experiments.

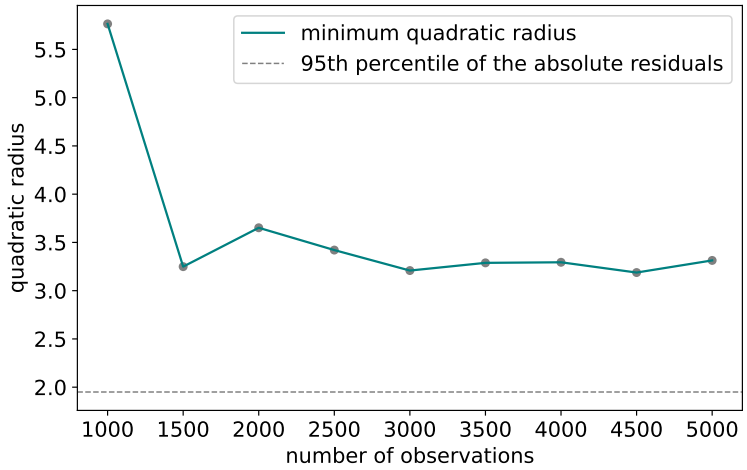


Figure 6 Minimal Radius of inferred quadratic region of the quantile Huber (solid line), compared to the 95% confidence region for residuals (dashed line), when errors are Gaussian with standard deviation 1. In every experiment, the quantile Huber framework infers a large κ value that pushes all residuals into the inlier region.

For the quantile Huber experiment, we generate asymmetric errors from the quantile Huber distribution with parameters $\tau^* = 0.1$ and $\kappa^* = 1.0$. Just as in the Gaussian case, we plot the relative error in recovery of x across 10 realizations for each experiment. The results are shown in Figure 7. The performance of the quantile Huber is far superior to that of the least squares better than that of LAD solutions. The inferred shape parameters compared to ground truth are show in Figure 8. As the number of observations increases, the ability to infer shape parameters also improves.

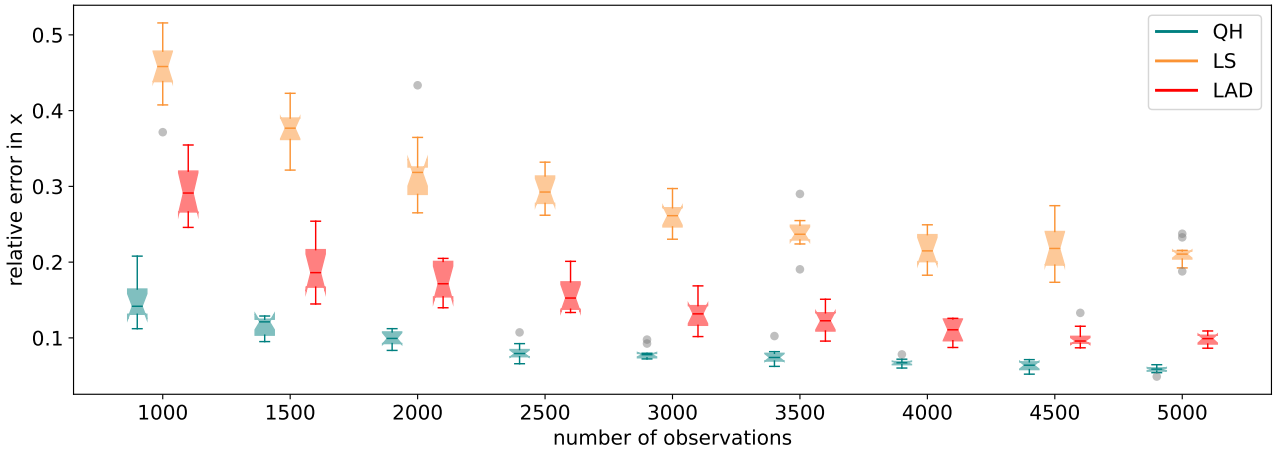


Figure 7 Relative error in recovery of regression parameters x for quantile huber (QH), least squares (LS) and least absolute deviation (LAD) approaches when measurement errors come from the quantile Huber ($\tau^* u = 0.1$, $\kappa^* = 1.0$). Box plots summarize results across 10 realizations of each experiment. The quality of quantile Huber result is far superior to those of both the least squares and LAD solutions across all observations. The least squares solution is particularly vulnerable to outliers, but the LAD solution is also affected by the asymmetry of errors.

5.3 Shape-Optimized Quantile Huber vs. Least Square and LAD

In this section, we stay with problem size $n = 50$ and $m = 1000$, and compare performance for errors generated from different quantile Huber distributions. The measurement errors are sampled from quantile Huber distributions, to verify that the approach is able to recover "ground truth" values for (τ, κ) parameters. We denote ground truth parameters as x_t, τ_t, κ_t, while x^*, τ^*, κ^* are the solutions obtained by solving (7). We provide two reference solutions: x_{LS} is the least square solution, and x_M is the solution obtained by solving

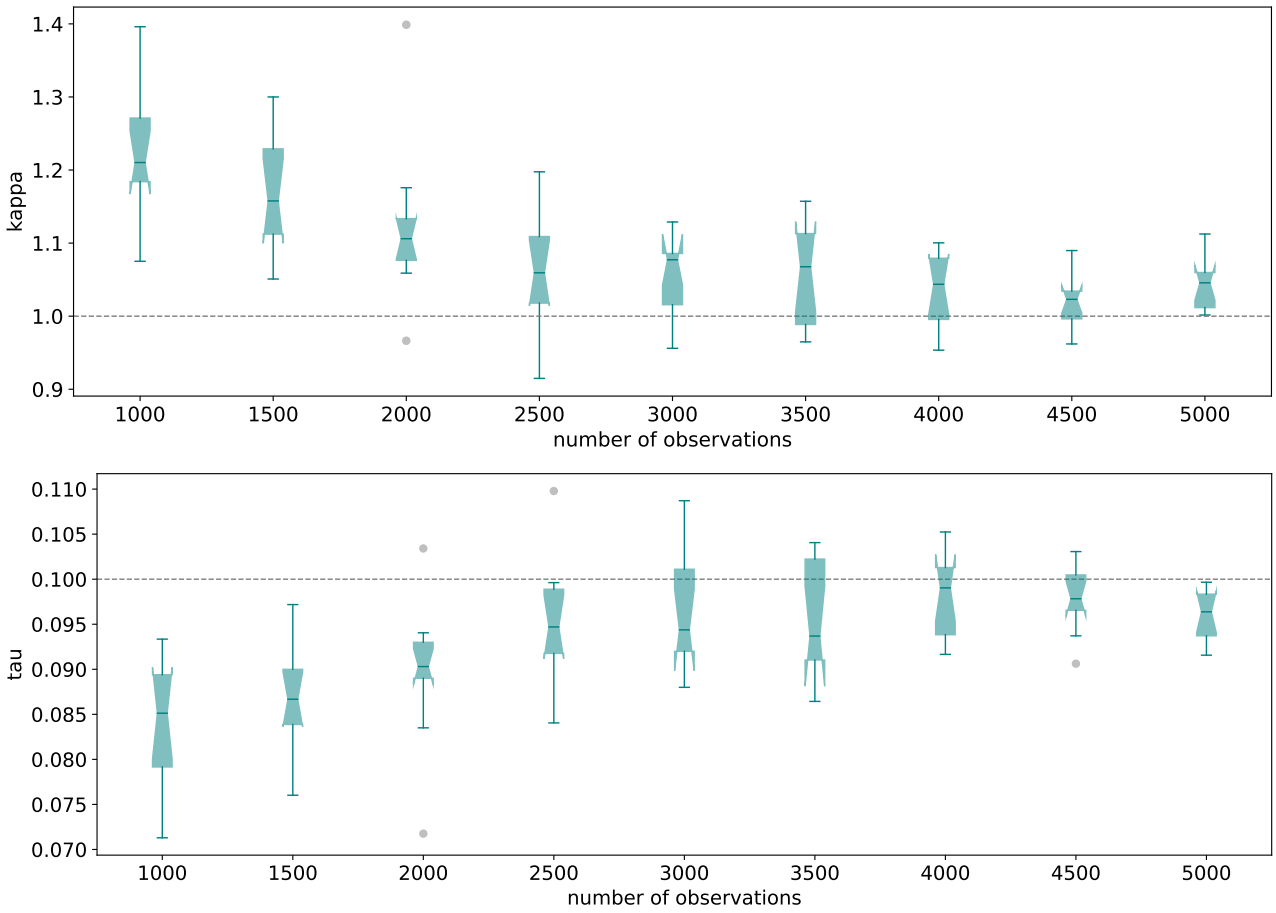


Figure 8 Estimates of τ (top panel) and κ (bottom panel) compared to ground true values $\tau^* = 0.1$ and $\kappa^* = 1.0$, shown using dashed lines in both panels. Shape parameters are estimated reasonably well, with less bias evident as the number of observations increases.

$\|Ax - b\|_1$ using `convex.jl`. For each κ and τ setting, we run the simulation 10 times, and show the average of the results in Table 2. Results shown are obtained by the IP method.

Table 2 Joint inference of the shape and model parameters for the quantile Huber loss. Columns 2 and 3 show results for proposed method. $r(x) = \|x - x_t\|/\|x_t\|$ denotes relative error. Column 2 contains τ, κ estimated using joint optimization (compare to ground truth simulation parameters in column 1) solved by the proposed IP algorithm with KKT tolerance set to 10^{-6}. Column 3 shows relative error of the new estimate; compare to Columns 4 and 5, which are relative errors for LS and minimum 1-norm estimates.

$[\tau_t, \kappa_t]$	$[\tau^*, \kappa^*]$	$r(x^*)$	$\text{sd}(r(x^*))$	$r(x_{LS})$	$\text{sd}(r(x_{LS}))$	$r(x_M)$	$\text{sd}(r(x_M))$
[0.1,1.0]	**[0.09,1.17]**	**0.14**	0.017	0.41	0.040	0.26	0.044
[0.2,1.0]	**[0.20,1.07]**	**0.10**	0.020	0.16	0.034	0.13	0.028
[0.5,1.0]	**[0.50,0.95]**	**0.08**	0.015	0.12	0.013	0.09	0.010
[0.8,1.0]	**[0.81,1.04]**	**0.09**	0.009	0.19	0.031	0.11	0.025
[0.9,1.0]	**[0.91,1.17]**	**0.12**	0.022	0.38	0.049	0.27	0.026

The maximum likelihood formulation correctly recovers the shape parameters (θ, τ). Moreover, the solution x^* obtained from the self-tuned regression is always better compared to reference solutions, and the improvement increases as measurement errors become more biased (τ close to 0 or to 1).

5.4 PALM vs. IP for Penalty Optimization

We also compared the performance of PALM and IP over $m \in \{100, 500, 1000, 2000\}$ and $n \in \{50, 100, 200, 500\}$, for both iterations and run time. The results are shown in Figure 9 and Table 3. From Table 3, IP converges

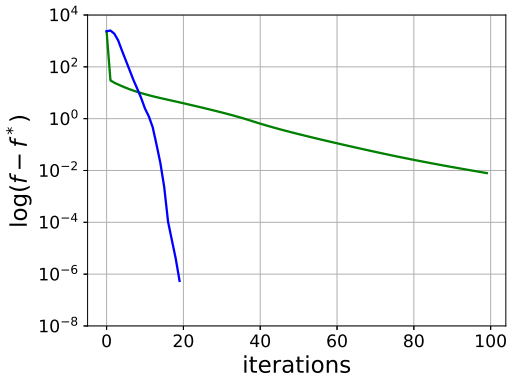


Figure 9 Convergence history (iterations) for PALM (green) and interior point method (black). The experiment shown is for $\tau = 0.1, \kappa = 1$. This behavior is representative; in particular the proposed IP method converges in fewer than 20 iterations in all cases.

Table 3 Timing comparison (in seconds) of PALM and IP for the quantile Huber family. Columns 2 and 3 show total run time and number of iterations of PALM and IP; column 3 plots difference in final objective values. IP finds lower objectives, but PALM is faster for larger problems. A large-scale implementation of IP that uses iterative methods in each iteration would help with the scaling issues.

m	n	PALM: Time(s)/#iter	IP: Time(s)/#iter	$f^*_{\text{PALM}} - f^*_{\text{IP}}$
100	50	4.12/96	3.11/15	7.25e-12
500	50	5.24/197	4.86/16	1.38e-08
1000	50	5.97/112	11.68/14	1.52e-08
2000	100	11.41/129	69.85/16	1.30e-08
2000	200	22.72/225	72.89/16	9.61e-08
2000	500	66.30/394	87.85/18	3.86e-08

within 20 iterations independently of problem dimension. However, the total run time exceeds PALM as the problem size increases. The current interior point is implemented by solving explicit linear systems, and is dominated by the cost of forming and solving linear systems. For our problems, with $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ with $m \geq n$, the cost of forming and solving T_1 (see Theorem 7) is $O(mn^2 + n^3)$, and it is updated in every iteration. In contrast, PALM uses linearization, so the cost of each iteration is proportional to that of a matrix-vector multiply, which is $O(mn)$ in this case. Each iteration is thus faster, but PALM needs more of them. The balance between IP and PALM can be shifted in favor of IP by implementing iterative solvers with pre-conditioners; this would be particularly valuable for nonsmooth ρ, where PALM cannot be applied. However, this effort is outside the scope of the paper.

Table 4 Column 3 shows time (s)/iterations for gradient descent on a single instance of the quantile huber problem. Columns 4 and 5 show the time/iterations for PALM for the joint problem over (x, θ) and resulting estimates of (κ, τ). Columns 6 and 7 shows the time and estimates obtained by a 5×5 grid search over (κ, τ). Penalty optimization finds good estimates very quickly relative to the baseline. Grid search takes an order of magnitude more time.

m	n	x only: Time (s)/#iter	Joint: Time(s)/#iter	$[\kappa^*, \tau^*]$	Cross-Val Time(s)	$[\kappa_{CV}, \tau_{CV}]$
100	50	24.57/4008	16.98/2198	[0.50,1.28]	585.12	[0.5,1.5]
500	50	4.13/430	7.31/386	[0.53,1.03]	309.96	[0.5,1.0]
1000	50	3.78/202	7.58/224	[0.49,1.00]	349.00	[0.5,1.0]
2000	100	14.22/219	19.79/255	[0.49,1.02]	1026.92	[0.5,1.0]
2000	200	29.20/388	42.67/513	[0.49,1.04]	2315.02	[0.5,1.0]
2000	500	158.32/1025	202.23/1261	[0.52,1.20]	7477.29	[0.5,1.0]

5.5 Penalty Optimization vs. Grid Search

Finally, we compared the run time and shape parameter estimates for the quantile Huber family with a grid search method. We construct a 5×5 grid over $\tau \in [0.1, 0.2, 0.5, 0.8, 0.9]$ and $\kappa \in [0.1, 0.5, 1.0, 1.5, 1.9]$, split the data into training and validation sets (80% and 20%) and pick the parameter that performs the best over validation data. We also provide the run time for a single instance evaluation with fixed θ (using gradient descent) as a baseline comparison.

Results are shown in Table 4. From the table we can see that both methods find good estimates of the shape parameters. However, grid search clearly suffers form the "curse of dimensionality" even at dimensionality 2, as it takes approximately 20 times longer to complete.

6 Real Data Example

In this section, we consider large-scale examples, developing approaches for using "self-tuning" penalties for robust principal component analysis (RPCA). RPCA has applications to alignment of occluded images [26], scene triangulation [34], model selection [14], face recognition [31] and document indexing [13]. We develop a self-tuning background separation approach. Given a sequence of images [23], our goal is to separate the moving objects from the background. We pick 202 images from the data set, convert them to grey scale and reshape them as column vectors of matrix $Y \in \mathbb{R}^{20480\times 202}$. We model the data Y as the sum of low rank component L and sparse noise S; we expect moving objects to be captured by S.

We take advantage of the fact that RPCA is equivalent to regularized Huber regression [16]:

$$\min_{L,S} \frac{1}{2\sigma^2}\|L+S-Y\|_F^2 + \kappa\|S\|_1 + \lambda\|L\|_* = \min_{L} \rho(Y-L;[\kappa;\sigma]) + \lambda\|L\|_*. \tag{20}$$

where $\rho(\,\cdot\,,[\kappa;\sigma])$ is the scaled Huber function given by

$$\rho(r;[\kappa,\sigma]) = \begin{cases} \kappa|r|/\sigma - \kappa^2/2, & |r| > \kappa\sigma \\ r^2/(2\sigma^2), & |r| \le \kappa\sigma. \end{cases}$$

When the variance σ^2 of the observation noise is known, the problem reduces to Huber threshold estimation; when it is not known, it is an additional shape parameter we can estimate using the proposed framework. In the latter case, there is no simple parametrization that makes the PLQ representation an affine function of θ.

To obtain a more efficient numerical scheme, we can model the low rank component as $L = U^\mathsf{T}V$, where $U \in \mathbb{R}^{k\times m}$ and $V \in \mathbb{R}^{k\times n}$. The resulting self-tuning RPCA formulation is given by

$$\min_{U,V,\kappa>0,\sigma>0} \sum_{i,j} \rho(\langle U_i, V_j\rangle - Y_{i,j};[\kappa,\sigma]) + mn\log[n_c([\kappa,\sigma])].$$

We solve the problem with PALM, obtaining the result in Figure 10(b). As the optimization proceeds, κ and σ decrease to 0 with a fixed ratio $\alpha = \kappa/\sigma$. The self-tuning Huber becomes the scaled 1-norm, recovering the original RPCA formulation [13]. The result in Figure 10(b) is an improvement over the result with initial (κ,σ) values shown in Figure 10(a).

Table 5 Runtime comparison for self-tuning Huber RPCA (Joint Eestimation) v.s. single Huber RPCA with fixed (κ,σ) parameters. We fix number of iterations to be 200. Joint estimation takes less than 2× run time for solving an RPCA problem with fixed parameters.

Joint Estimation: Time (s)/#iter	Fixed Parameters: Time (s)/#iter
203.11/200	126.72/200

As in Section 5, we compared the run time of solving the joint (self-tuning) problem with solving a single instance of RPCA. The results are shown in Table 5. In this case, the extended self-tuning problem only takes twice the run time compared to a single instance. Cross-validation, grid-search and black-box optimization typically require multiple solutions of the original optimization problem to find good parameter values.

(a) Huber with $\kappa = 0.002, \sigma = 1$ **(b)** Self-tuned Huber, initial: $\kappa = 0.002$, $\sigma = 1$

Figure 10 RPCA background separation: we optimize over parameters as well as the foreground and background. The separation is better as a result of the joint optimization.

7 Discussion

In this paper we developed a simple approach that extends a regression problem using PLQ penalties to also infer unknown shape constraints. We used the statistical interpretation of the PLQ penalty to introduce an additional term that relates shape constraints to a normalization constant.

Many existing algorithms can be brought to bear on the extended problem. For smooth penalties, the PALM algorithm was quite useful in both synthetic and real examples, particularly for large-scale problems. We also developed an interior point method for the PLQ class, that uses conjugate representations of such penalties to solve an augmented saddle point system.

The approach offers several interesting avenues for future research. The nonconvex coupling between convex PLQ penalties and their shape parameters is interesting, and finding conditions when the approach is guaranteed to work in general is an open question. The maximum likelihood criterion itself is just one approach, and may have limitations as the number of shape parameters grows with respect to the data. Characterizing these limitations and trying alternatives (such as marginal likelihood) is also left to future work.

References

1 Aleksandr Y. Aravkin, James V. Burke, and Gianluigi Pillonetto. Sparse/Robust Estimation and Kalman Smoothing with Nonsmooth Log-Concave Densities: Modeling, Computation, and Theory. *J. Mach. Learn. Res.*, 14:2689–2728, 2013.

2 Aleksandr Y. Aravkin, James V. Burke, and Gianluigi Pillonetto. Generalized system identification with stable spline kernels. *SIAM J. Sci. Comput.*, 40(5):B1419–B1443, 2018.

3 Aleksandr Y. Aravkin, Dmitriy Drusvyatskiy, and Tristan van Leeuwen. Efficient quadratic penalization through the partial minimization technique. *IEEE Trans. Autom. Control*, 63(7):2131–2138, 2017.

4 Aleksandr Y. Aravkin and Tristan Van Leeuwen. Estimating nuisance parameters in inverse problems. *Inverse Probl.*, 28(11), 2012.

5 Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Stat. Surv.*, 4:40–79, 2010.

6 Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka–Łojasiewicz inequality. *Math. Oper. Res.*, 35(2):438–457, 2010.

7 Bradley M. Bell, James V. Burke, and Alan Schumitzky. A relative weighting method for estimating parameters and variances in multiple data sets. *Comput. Stat. Data Anal.*, 22(2):119–135, 1996.

8 Anil K. Bera, Antonio F. Galvao, Gabriel V. Montes-Rojas, and Sung Y. Park. Asymmetric Laplace Regression: Maximum Likelihood, Maximum Entropy and Quantile Regression. *J. Econom. Methods*, 5(1):79–101, 2016.

9 James S. Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, pages 2546–2554. MIT Press, 2011.

10 James S. Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:281–305, 2012.

11 Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program.*, 146(1-2):459–494, 2014.

12 Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

13 Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3), 2011.

14 Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Sparse and low-rank matrix decompositions. *IFAC Proceedings Volumes*, 42(10):1493–1498, 2009.

15 David L Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.

16 Derek Driggs, Stephen Becker, and Aleksandr Y. Aravkin. Adapting regularized low-rank models for parallel architectures. *SIAM J. Sci. Comput.*, 41(1):A163–A189, 2019.

17 Peter J. Huber. *Robust statistics*, volume 523. John Wiley & Sons, 2004.

18 Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International Conference on Learning and Intelligent Optimization*, pages 507–523. Springer, 2011.

19 Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, and Frank Hutter. Fast Laplace optimization of machine learning hyperparameters on large datasets. `https://arxiv.org/abs/1605.07079`, 2016.

20 Roger Koenker and Kevin F. Hallock. Quantile regression. *J. Econom. Perspect.*, 15(4):143–156, 2001.

21 Masakazu Kojima, Nimrod Megiddo, Toshihito Noma, and Akiko Yoshise. *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*, volume 538 of *Lecture Notes in Computer Science*. Springer, 1991.

22 Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. `https://arxiv.org/abs/1603.06560`, 2016.

23 Liyuan Li, Weimin Huang, Irene Yu-Hua Gu, and Qi Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Trans. Image Process.*, 13(11):1459–1472, 2004.

24 Arkadi Nemirovski and Yurii Nesterov. *Interior-Point Polynomial Algorithms in Convex Programming*, volume 13 of *Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics, 1994.

25 Dominique Orban and Mario Arioli. *Iterative Solution of Symmetric Quasi-Definite Linear Systems*, volume 3 of *SIAM Spotlights*. Society for Industrial and Applied Mathematics, 2017.

26 Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2233–2246, 2012.

27 R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational analysis*, volume 317. Springer, 2009.

28 Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959. Curran Associates Inc., 2012.

29 Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc., Ser. B, Stat. Methodol.*, 58(1):267–288, 1996.

30 Shiyi Tu, Min Wang, and Xiaoqian Sun. Bayesian variable selection and estimation in maximum entropy quantile regression. *J. Appl. Stat.*, 44(2):253–269, 2017.

31 Matthew A. Turk and Alex P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition*, pages 586–591. IEEE, 1991.

32 Stephen J. Wright. *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics, 1997.

33 Keming Yu and Rana A. Moyeed. Bayesian quantile regression. *Stat. Probab. Lett.*, 54(4):437–447, 2001.

34 Zhengdong Zhang, Arvind Ganesh, Xiao Liang, and Yi Ma. Tilt: Transform invariant low-rank textures. *Int. J. Comput. Vision*, 99(1):1–24, 2012.