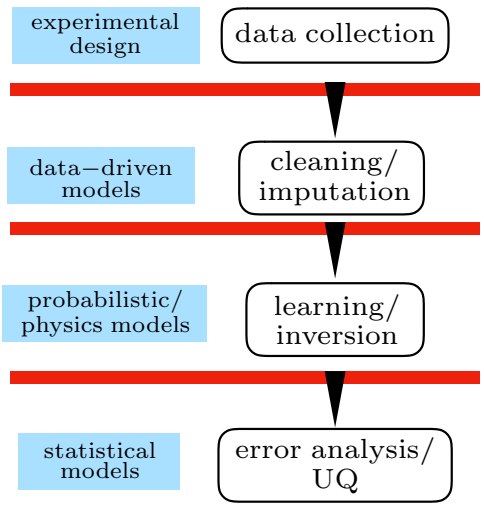
Unified optimization for self-learning robust penalties.

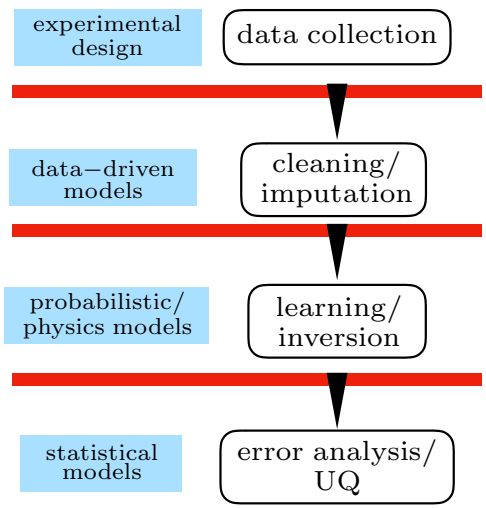
Aleksandr Aravkin[1]
Peng Zheng [2]

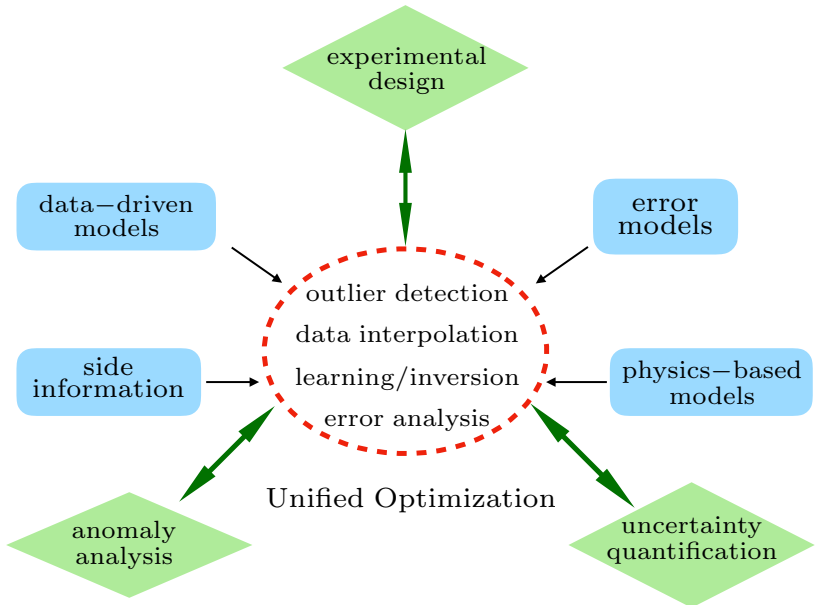
[1] Applied Math & eSciences, UW
[2] Applied Math, UW

Classic vs. unified approach to scientific discovery.



Classic vs. unified approach to scientific discovery.



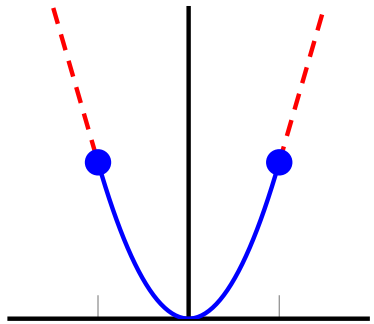


Inversion & Learning Problems

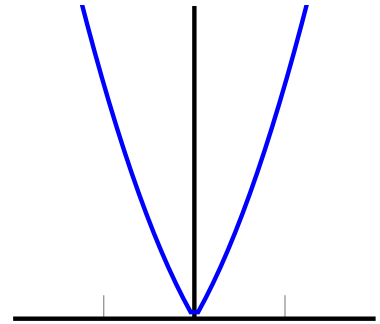
$$\begin{aligned} \min_{x} \quad & \rho_1(Ax - b) + \rho_2(Cx) \\ \text{s.t.} \quad & Fx \leq f. \end{aligned}$$

- Data misfit: good results in the face of *large measurement errors*
- Regularization: *prior information* e.g. sparsity.
- Constraints: use information about feasible region

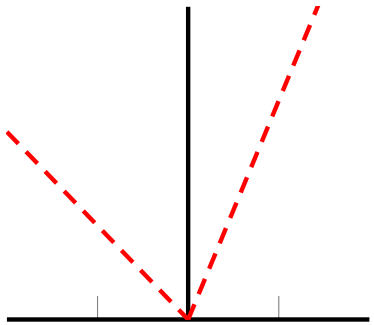
Penalties with Parameters



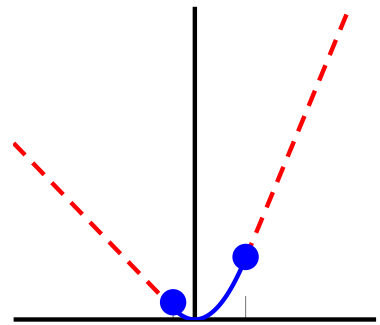
(a) huber, $\kappa = 1$



(b) elastic net, $\alpha = 0.5$



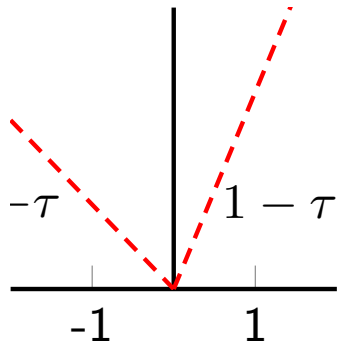
(c) quantile, $\tau = 0.3$

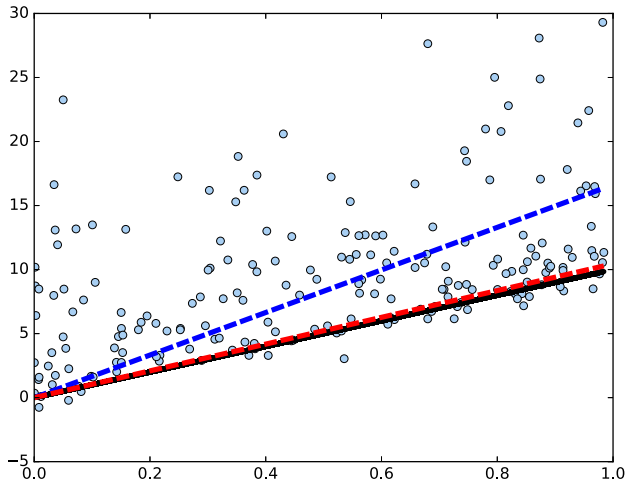


(d) quantile huber

Figure: Piecewise linear-quadratic penalties with shape parameters.

Central Theme: Learning the Learning Criterion





$$q(r;\tau) = (1-\tau)r_+ + \tau r_-.$$

$$\min_{\substack{x \\ \tau\in[0,1]}} \; q(Ax-b;\tau) \quad + \quad ???$$

Tuning Parameters

- Tuning parameters is difficult.

- Current methods:
  - cross-validation
  - random search
  - Bayesian optimization

- We can simultaneously solve for shape parameters and the model x.

Statistical View

We assume ϵ_i are drawn from density

$$p(r;\theta) = \frac{1}{n_c(\theta)} \exp[-\rho(r;\theta)], \quad n_c(\theta) = \int_{\mathbb{R}} \exp[-\rho(r;\theta)]\, dr$$

where θ controls the shape, and $n_c(\theta)$ is a normalization constant.

Maximum Likelihood

The joint maximum likelihood (in x and θ) is equivalent to:

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

Self-tuning quantile penalty formulation:

$$\min_{x,\tau \in [0,1]} \sum_{i=1}^{m} q(y_i - \langle a_i, x \rangle ; \tau) + m \log \left(\frac{1}{\tau} + \frac{1}{1-\tau} \right)$$

Self-Tuned Quantile Result

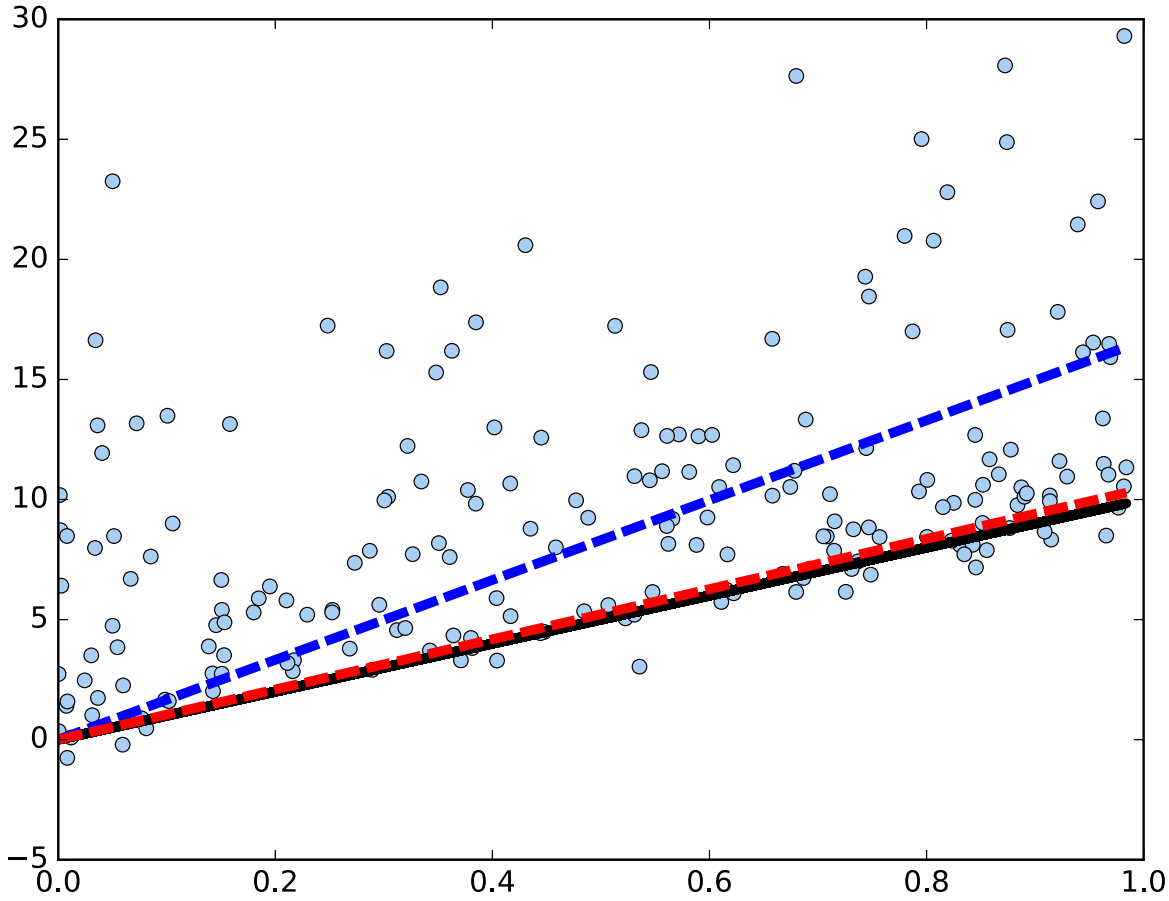


Figure: true generator (black), least squares (blue), self-tuned quantile huber (red).

Restricting the Problem Class

Assumption

Consider $\rho(r;\theta) : \mathcal{D} \times \mathbb{R} \to \mathbb{R}$, *such that*

1. $\rho(r;\theta) \geq 0$ *for any* $r \in \mathbb{R}$ *and* $\theta \in \mathcal{D}$.
2. *For any* $\theta \in \mathcal{D}$, $n_c(\theta) = \int_{\mathbb{R}} \exp[-\rho(r;\theta)]\, dr < \infty$.
3. *For any* $\theta_0 \in \mathcal{D}$, $\rho(r;\theta)$ *is* $\mathcal{C}^2$ *around* θ_0 *for almost every* $r \in \mathbb{R}$.

Smoothness

Theorem (smoothness of $n_c(\theta)$)

For $n_c(\theta) = \int_{\mathbb{R}} \exp[-\rho(r;\theta)]\, dr$, if Assumption holds and for any $\theta_0 \in \mathcal{D}$, there exist functions $g_k(r)$, $k = 1, 2$, such that,

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

then $n_c(\theta)$ is $\mathcal{C}^2$ in nbhd of θ_0 and,

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

Examples

- Piecewise linear quadratic (PLQ) penalties, e.g. huber, quantile, quantile-huber, ℓ_2, ℓ_1 and elastic net.
- Other robust penalties, e.g. negative log-likelihood of Student's T:

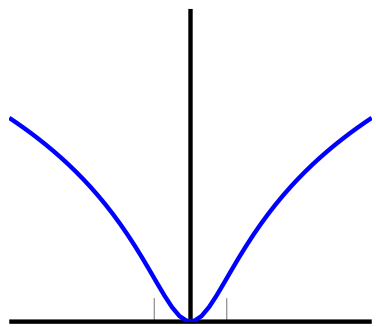


Figure: Student's T: $\rho(r; v) = \log(1 + x^2/v)$

Convexity

Theorem (convexity of $\log(n_c(\theta))$)

Suppose Assumption holds, and let $\rho(r;\theta)$ be a coercive convex non-negative function of r for every θ. We have the following results:

1. *If $\rho(r;\theta)$ is 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.*

Bottom Line

The joint optimization problem is *never convex*. However, in many cases local search finds global optima.

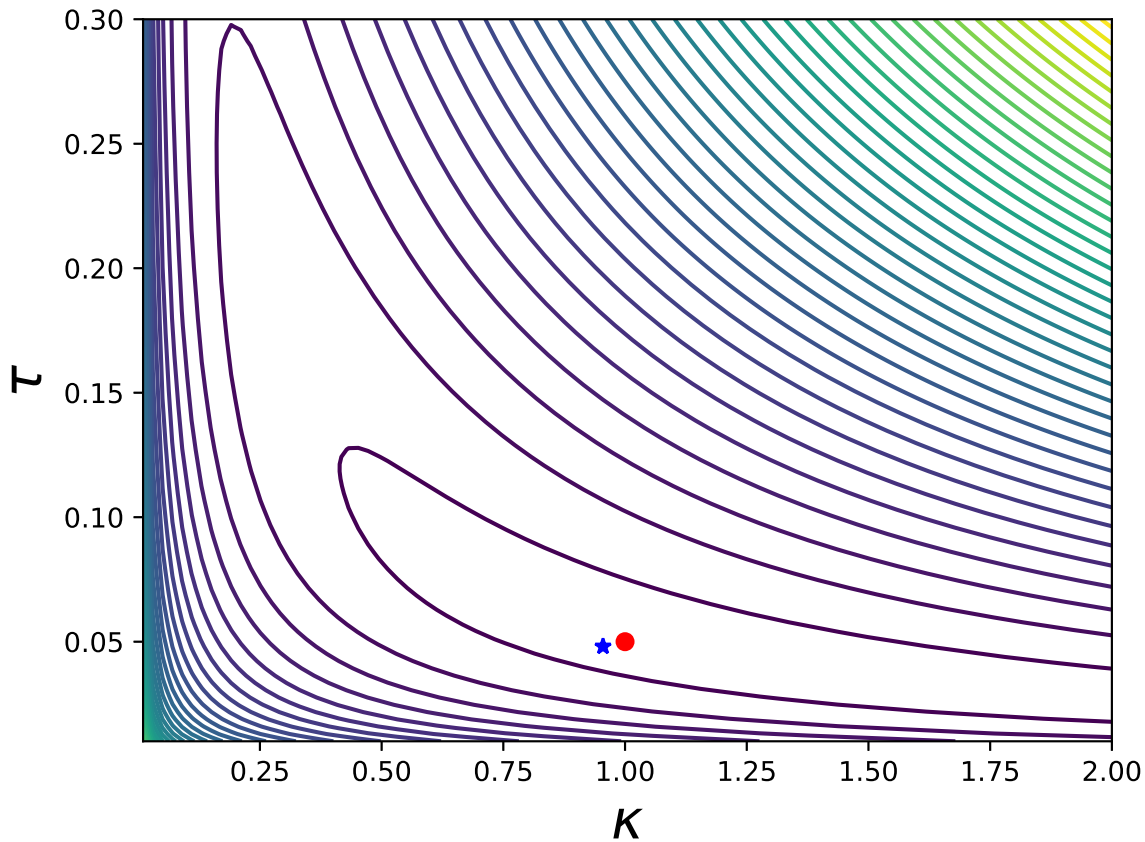
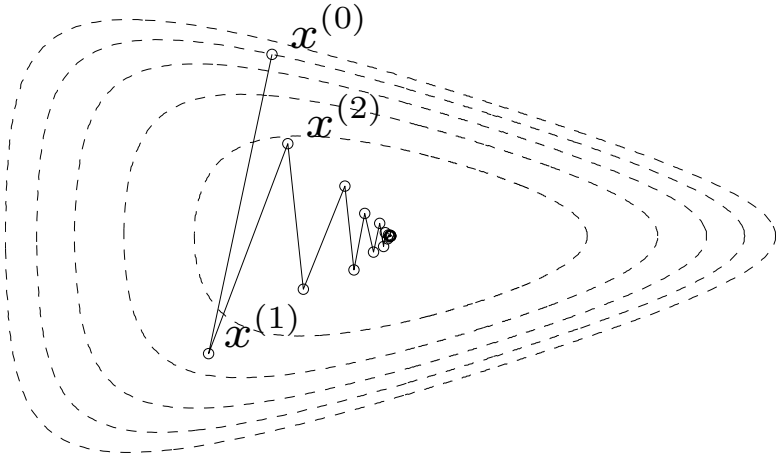
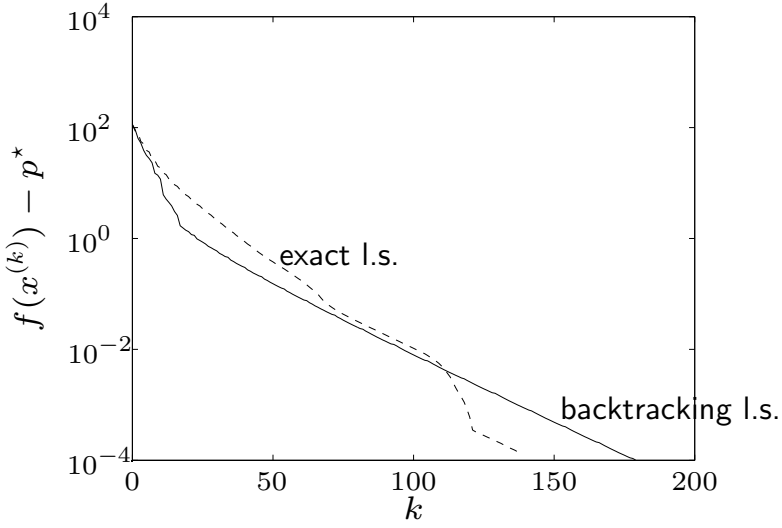
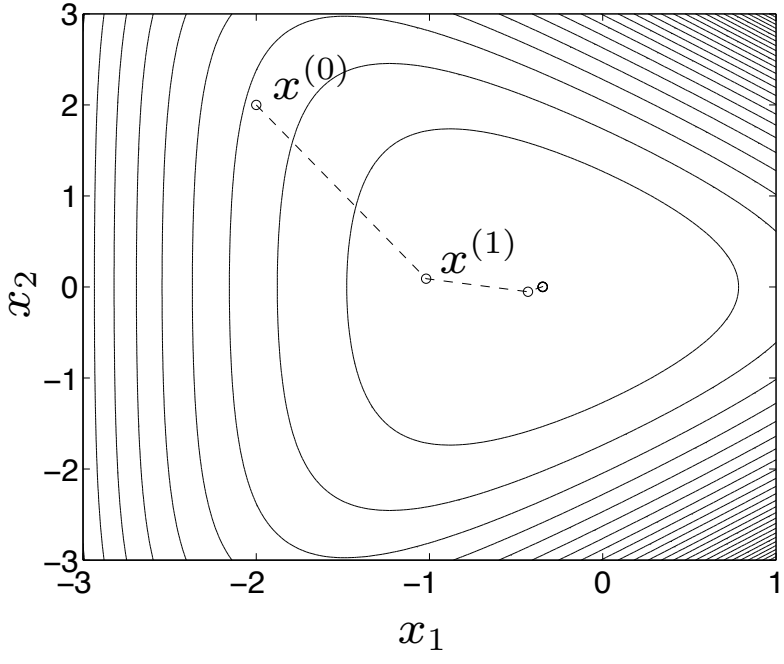


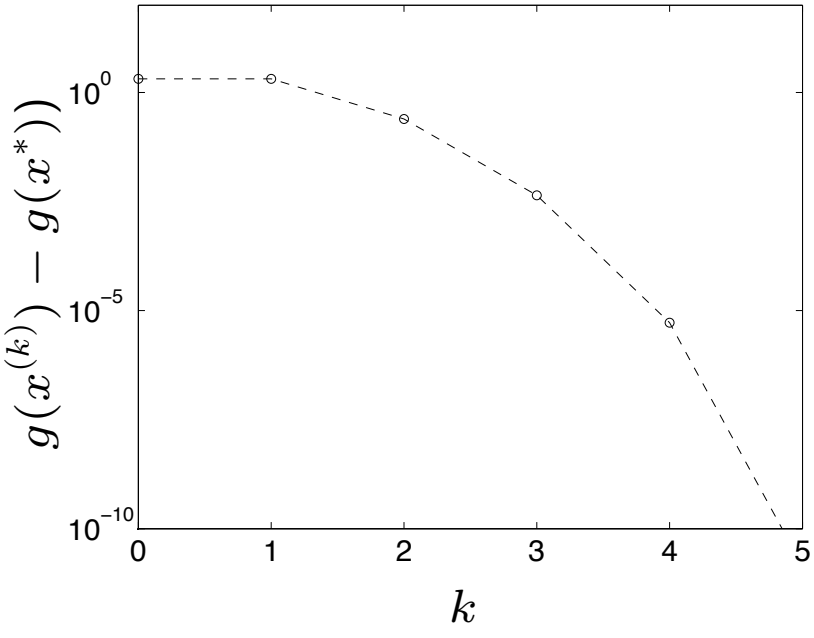
Figure: Level sets for value function $v(\theta) := \min_x \rho(x, \theta)$ for the quantile Huber model. The blue star is the maximum likelihood estimator, while the red dot represents the true parameters in the simulation.

First and second order methods









PALM

To apply PALM[3] to the joint maximum likelihood problem for smooth ρ, take

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

$$r_1(x) = g(x), \quad r_2(\theta) = \delta_{\mathcal{D}}(\theta) + m\log[n_c(\theta)].$$

where where δ is indicator function and g is'prox-friendly' regularizer.

Algorithm 1 PALM

Input: A, y

Initialize: x^0, θ^0

1: **while** not converge **do**

2: $\quad x^{k+1} \leftarrow \text{prox}_{c_k^{-1} r_1}\left[x^k - c_k^{-1}\nabla_x H(x^k,\theta^k)\right]$

3: $\quad \theta^{k+1} \leftarrow \text{prox}_{d_k^{-1} r_2}\left[\theta^k - d_k^{-1}\nabla_\theta H(x^{k+1},\theta^k)\right]$

Output: x^{k+1}, θ^{k+1}

[3] Bolte, Sabach & Teboulle, 2014

PALM Pros and Cons

Pros:

- Flexibility.
- Low per-iteration cost (we use it for larger examples).

Cons:

- Requires smooth H (so quantile is out.)
- Needs many iterations (first-order method).

PLQ Penalties

All PLQ penalties have a simple dual representation:

- Quantile:

$$q(r;\tau) = \sum_{i=1}^{m} \begin{cases} -\tau r_i, & r_i < 0 \\ (1-\tau) r_i, & r_i \geq 0 \end{cases} = \sup_{u \in [-\tau, (1-\tau)]^m} \left\{ u^\mathsf{T} r \right\}$$

- Huber:

$$h(r;\kappa) = \sum_{i=1}^{m} \begin{cases} \kappa |r_i| - \kappa^2/2, & |r_i| > \kappa \\ r^2/2, & |r_i| < \kappa \end{cases} = \sup_{u \in [-\kappa, \kappa]^m} \left\{ u^\mathsf{T} r - \frac{1}{2} u^\mathsf{T} M u \right\}$$

- Quantile-Huber:

$$qh(r;[\tau,\kappa]) = \sup_{u \in [-\tau\kappa, (1-\tau)\kappa]^m} \left\{ u^\mathsf{T} r - \frac{1}{2} u^\mathsf{T} M u \right\}$$

PLQ Class

PLQ class with explicit shape parameters θ:

$$\rho(r; B, \bar{b}_\theta, C, \bar{c}_\theta, M) = \sup_u \left\{ u^\mathsf{T}(Br - \bar{b}_\theta) - \frac{1}{2} u^\mathsf{T} M u \mid C^\mathsf{T} u \leq \bar{c}_\theta \right\}$$ [4], [5]

$$\bar{b}_\theta = G^\mathsf{T}\theta + b, \quad \bar{c}_\theta = H^\mathsf{T}\theta + c$$ [6]

[4] Rockafellar and Wets, Variational Analysis, 2009.

[5] A., Burke, and Pillonetto. "Sparse/robust estimation and kalman smoothing with nonsmooth log-concave densities: Modeling, computation, and theory." The Journal of Machine Learning Research 14.1 (2013): 2689-2728.

[6] Zheng, P., A., and Ramamurthy, K. "Shape Parameter Estimation." arXiv preprint arXiv:1706.01865 (2017).

Interior Point Approach

Our full problem (in x, θ) is given by

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

- x are (original) regression variables.
- u are 'conjugate' variables from PLQ representation.
- θ encode shape parameters.

Overview of Interior Point:

- $z := (x, u, \theta)$.
- Replace constraints using logarithmic barrier, e.g. $x \geq 0$ with $-\mu \ln(x)$.
- Define relaxed KKT system $F_\mu(z)$.
- solve $F_\mu(z) = 0$ using Newton while taking μ to 0.

KKT System

And our KKT system could be written as,

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

And the Jacobian matrix of the system is,

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

IP Pros and Cons

Pros:

- Solves PLQ problems and estimates PLQ shape parameters.
- Has a superlinear local convergence rate

Cons:

- Restricted to PLQ functions (for now).
- High per-iteration cost for large scale problems (for now).

Speed Comparison

Convergence rates for self-tuning Quantile Huber:

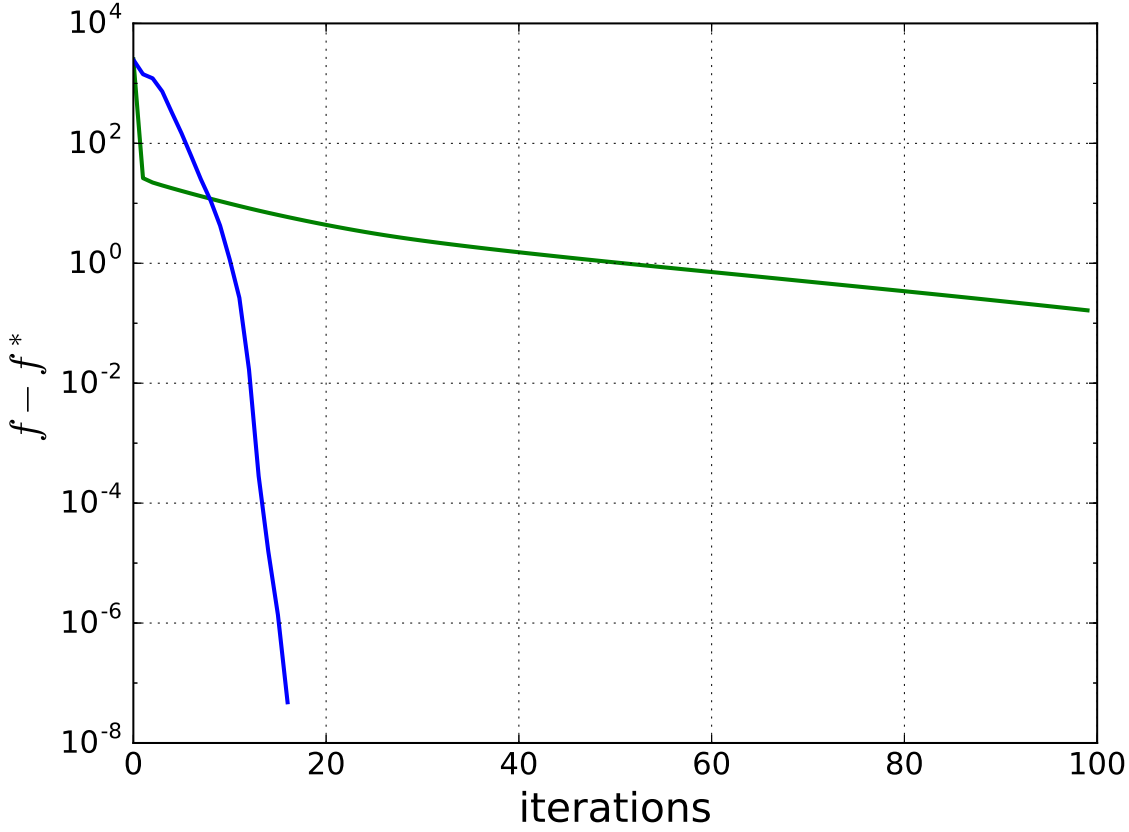


Figure: PALM: linear rate (green); IPsolve: super linear rate (blue).

Timing Comparison

Synthetic Data:

- ϵ_i i.i.d. quantile huber errors
- $A = \texttt{randn(m,n)}$, $x_t = \texttt{randn(n)}$, $y = Ax + \epsilon$

m	n	PALM: T/#iter	IP: T/#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

Table: Timing comparison (sec) of PALM and IP for the quantile Huber family. C2, 3 show total run time and number of iterations of PALM and IP; C3 plots difference in final objective values. IP finds lower values; PALM is faster for larger problems.

Quality of results

$[\tau_t, \kappa_t]$	$[\tau^*, \kappa^*]$	$r(x^*)$	$r(x_{LS})$	$r(x_M)$
[0.1,1.0]	**[0.09,1.17]**	**0.14**	0.41	0.26
[0.2,1.0]	**[0.20,1.07]**	**0.10**	0.16	0.13
[0.5,1.0]	**[0.50,0.95]**	**0.08**	0.12	0.09
[0.8,1.0]	**[0.81,1.04]**	**0.09**	0.19	0.11
[0.9,1.0]	**[0.91,1.17]**	**0.12**	0.38	0.36

Table: Joint inference of the shape and model parameters for quantile Huber regression. $r(x) = \|x - x_t\|/\|x_t\|$ denotes relative error. C2 contains τ, κ estimated using joint optimization (compare to C1). C3 shows relative error of the new estimate; compare to C4, 5 which are relative errors for LS and 1-norm estimates.

RPCA

Consider the foreground/background separation problem.
We want to separate background (low rank L) from moving objects in foreground (sparse S).
The dataset is built from 202 frames from a video clip, which are shaped into a matrix $Y \in \mathbb{R}^{20480 \times 202}$.

Inexact RPCA deconvolves S and L:

$$\min_{L,S} \frac{1}{2}\|L+S-Y\|_F^2+\kappa\|S\|_1+\lambda\|L\|_*$$

Huber in RPCA

RPCA is equivalent to a low rank Huber formulation:

$$\min_{U,V} \rho(U^\mathsf{T} V - Y; \kappa, \sigma)$$

$$U, V \quad \text{each have } k \text{ columns.}$$

where

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

Here, $\theta = (\kappa, \sigma^2)$.

Results for Self-Tuning Huber

Tuning κ through cross-validation is expensive, instead we could automatically tune κ by our approach.

Figure: Left: huber with fixed $\kappa = 2 \times 10^{-3}, \sigma = 1$ (INIT)
Right: self-tuned huber starting from INIT; final $\kappa = 1.94 \times 10^{-2}, \sigma = 8.28 \times 10^{-4}$

Huberized Student's T (Tiber)

We introduce a new penalty, huberized student's t (Tiber)[7]:

$$\rho(r;[\kappa,\sigma]) = \begin{cases} \frac{2\kappa}{\sigma(\kappa^2+1)}(|r| - \kappa\sigma) + \log(1+\kappa^2), & |r| > \kappa\sigma \\ \log(1 + r^2/\sigma^2), & |r| \leq \kappa\sigma \end{cases}$$

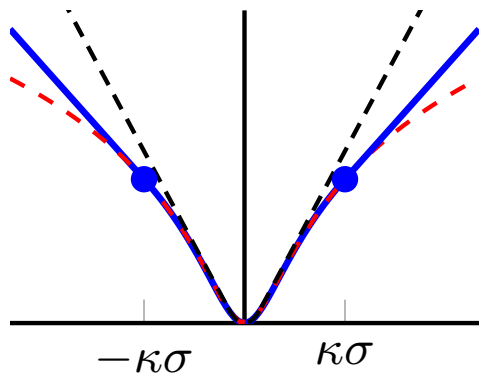


Figure: Huberized Student's t (thick blue) interpolates between Student's t (red dash) and Huber (black dash).

[7] Zheng, P., A., Ramamurthy, K., and Thiagarajan, J. "Learning Robust Representations for Computer Vision." RCL-ICCV, 2017. arXiv preprint arXiv:1708.00069.

Results for Self-Tuned Tiber

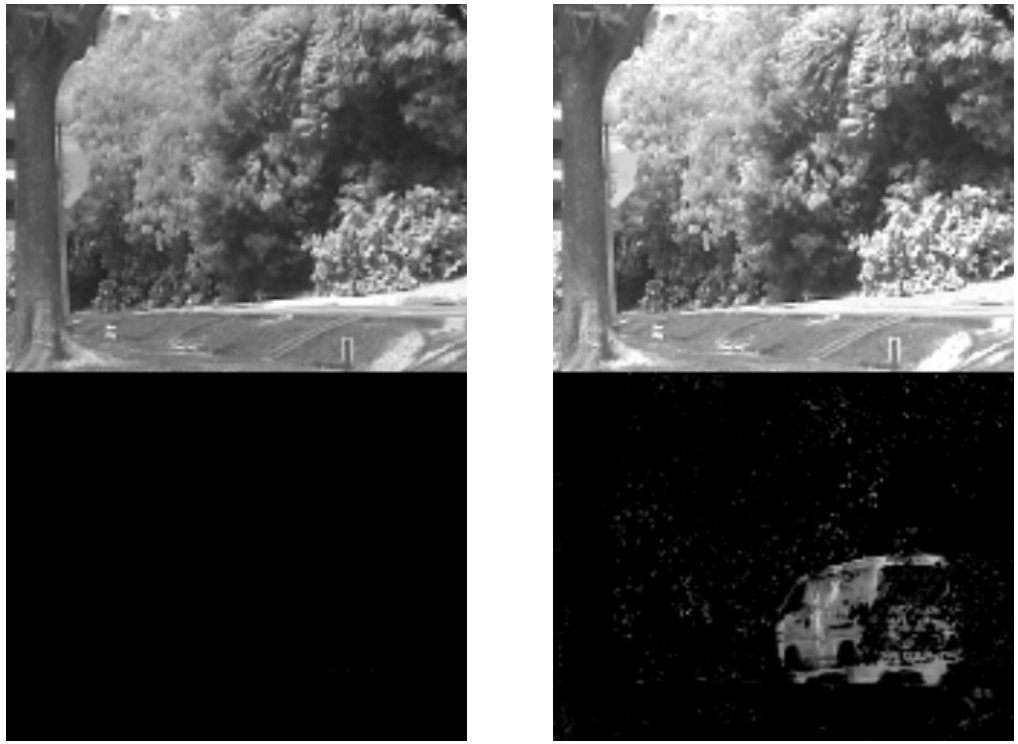
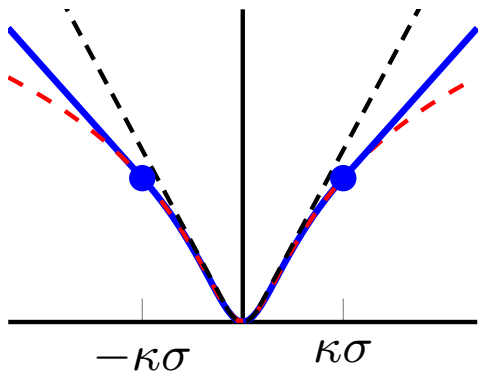


Figure: Left: Tiber with $\kappa = 2 \times 10^{-3}, \sigma = 1$ (INIT)
Right: Self-tuned Tiber from INIT; final: $\kappa = 7.64, \sigma = 2.24 \times 10^{-2}$

Self-Tuning Tiber RPCA: Escalator



(a) Tiber penalty (b) self-tuned Tiber RPCA[8]

$$\min_{U,V,\kappa>0,\sigma>0} \rho(UV^T - Y; [\kappa, \sigma]) + mn \log[n_c([\kappa, \sigma])].$$

[8] Zheng, Aravkin, Ramamurthy, "Shape parameter estimation". https://arxiv.org/abs/1706.01865

Conclusions

- Estimating densities for errors while fitting gives self-tuning formulations.
- Examples include Huber, quantile, elastic net, variance estimation.
- Simple examples have both real applications, and technical challenges.
- Q: Can we do shape constrained estimation for regression problems?