Robust Logistic Regression and Classification

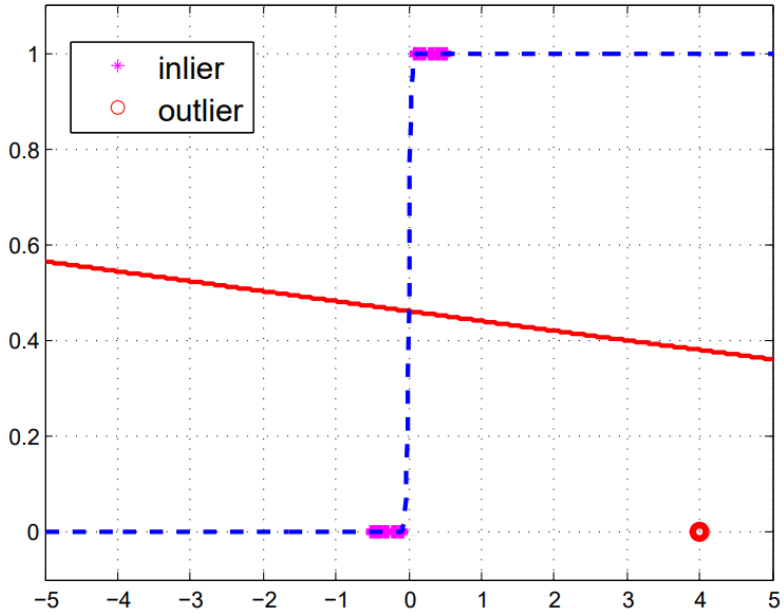
Jiashi Feng, Shie Mannor, Huan Xu, Suicheng Yan
2014

Background

Logistic Regression formulation:

$$P\{y_i = +1\} = \frac{1}{1 + exp(-\beta^\top x_i)}$$

Maximal-likelihood estimate of LR is very sensitive to outliers



Literature Review

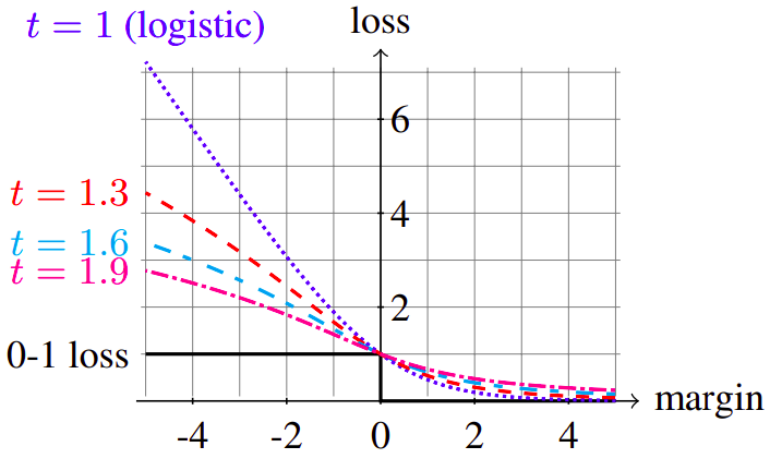
Common approach: M-estimator based, e.g. Pregiobon:

$$\hat{\beta} = \arg\min_{i=1} \sum_{i=1}^{n} \rho(\ell_i(\beta))$$

Huber-type function used for robustifying loss

Still sensitive to *high leverage covariates.*

T-logistic regression



Ding et. al, NIPS (2010)

Literature Review Cont.

Robust sparse regression:

- Use a trimmed standard vector inner product
- Convex programming method for estimating sparse parameters of logistic regression

$$\max_{\beta} \sum_{i=1}^{m} y_i \langle x_i, \beta \rangle, \text{ s.t. } \|\beta\|_1 \leq \sqrt{s}, \|\beta\| \leq 1$$

- But, this is still sensitive to higher counts of corrupted samples.

Problem Formulation: Logistic Regression

Assume an **uncorrupted** LR model:

$$P\{y_i = +1\} = \frac{1}{1 + \exp\left(-(\langle \beta^*, x_i \rangle + v_i)\right)}$$

With gaussian noise $v_i \sim \mathcal{N}(0, \sigma_e^2)$

A **constant number of samples may be adversarially corrupted**, without assumption: $n + n_1$ samples, n_1 can be corrupted.

Assumption: i.i.d. sub-Gaussian features

Main Takeaways

RoLR algorithm

1. Remove the samples with overly large magnitude
2. Maximize a trimmed correlation of remaining samples (using LP):

$$\hat{\beta} = \arg\max_{\beta \in B_2^p} \sum_{i=1}^{n} [y\langle \beta, x \rangle]_{(i)}$$

I.e. minimize a summation of top n inner products

Main Theorem

Theorem 1: Let λ be the ratio of the corrupoters to honest points: $\frac{n_1}{n}$, $\hat{\beta}$ be the output of RoLR, and β^* be the ground truth parameter. Then, we have with high probability ($> 1 - 4\exp(-\frac{c_2 n}{8})$) for some absolute constant c_2 and for subgaussian parameter p:

$$\left\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\right\| \leq 2\lambda A(\sigma_e^2, \sigma_x^2) + 2\mathrm{B}\left(\sqrt{\frac{p}{n}}\right) + 8\lambda\sigma_x^2\sqrt{\frac{\log p}{n} + \frac{\log n}{n}}$$

If we set look at the noiseless case $\sigma_e^2 = 1$, and set the feature variance $\sigma_x^2 = 1,$ and asymptotically $\frac{p}{n} \to 0$, then we get threshold

$$\left\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\right\| \lessapprox 3.54\lambda$$

Other details

Why does maximizing the correlation work? We want to minimize $\|\hat{\beta} - \beta^*\|$

$$\mathbb{E}\left[y\langle\beta, x\rangle - y\left\langle\beta', x\right\rangle\right] = \eta\left(1 - \langle\beta, \beta'\rangle\right) \geq \frac{\eta}{2}\|\beta - \beta'\|_2^2$$

Instead of a maximal likelihood estimation, reformulate to a linear programming problem:

$$\max_{\beta,\nu,\xi_i} -\nu \cdot n - \sum_{i=1}^{n+n_1} \xi_i, \text{ s.t. } y_i\langle\beta, x_i\rangle + \nu + \xi_i \geq 0, \beta \in B_2^p, \nu \geq 0, \xi_i \geq 0.$$

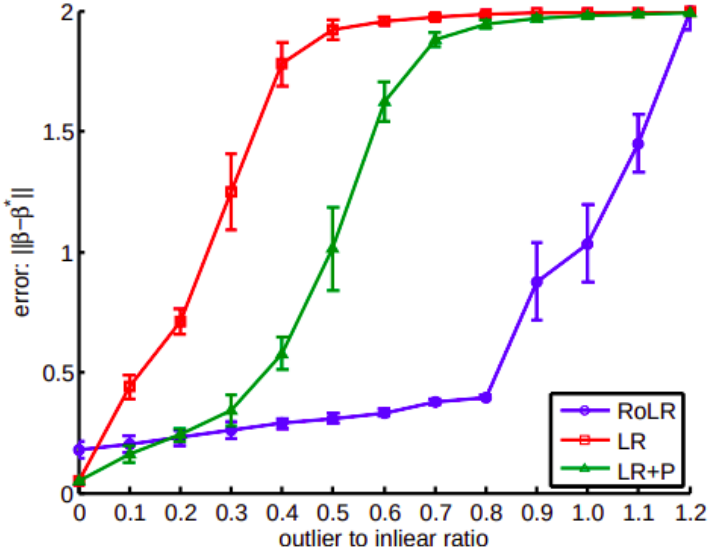
Other details: Binary Classification

If instead we are interested in deterministically labelled samples:

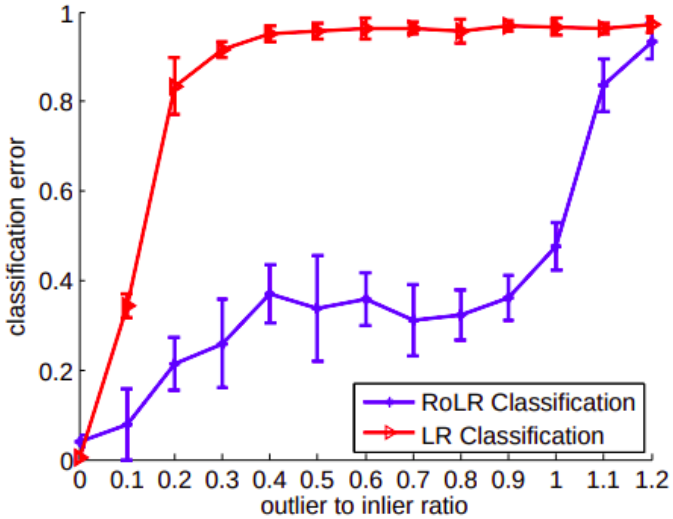
$$y_i = \operatorname{sign}\left(\langle \beta^*, x_i \rangle + v_i\right)$$

A similar argument follows from calculating the expectation of the correlation (the product $y_i \langle \beta, x_i \rangle$ and showing that minimizing this will bring us closer to the optimal parameter.

Simulation Results



(a) Logistic regression



(b) Classification

Discussion

Pros:

- Converges to global optimum
- Allow for any kind of corruptions. Bounds only depend on ratio of honest to corrupted samples, and the covariance of the signal and noise.
- Linear programming approach = better computational efficiency

Discussion cont.

Cons:

- Basic linear binary classification.
- In the purely honest samples situation: RoLR suffers some performance degradation.
- Sub-Gaussian requirement on generating the non-corrupted samples

Robust High-Dimensional Linear Regression

Chang Liu, Bo Li, Yevgeniy Vorobeychik, Alina Oprea
2017

Background

- Very closely related to last work. We now consider regression.
- Instead of arbitrarily corrupted rows, corruptions that deliberately try to mislead an algorithm trying to learn a low-dimensional representation of the data
- A note: PCR-based approaches are quite slow.

Problem Formulation

Assume that the feature matrix comes from an **adversarially-corrupted approximately low-rank matrix.**

Ground Truth	$\mathbf{y}_* = \mathbf{X}_* \beta^* = \mathbf{U} \beta_U^*$
True model	β^*
Low-dim representation	$\mathbf{U} = \mathbf{X}_* B$
Noisy feature matrix	$\mathbf{X}_0 = \mathbf{X}_* + \mathbf{N}$
Noise	$\text{variance } \sigma,\ \|\mathbf{N}\|_\infty \leq \epsilon; \mathbf{y}_0 = \mathbf{y}_* + e$
Corruption	Additional n1 rows of examples and labels

Main Takeaways

Goal: with high probability, being close to a model learned on the noiseless, uncorrupted classifier. Find estimate $\hat{\beta}$ of model parameter β.

1. Recover the subspace of the noiseless, uncorrupted classifier $\mathbf{X}_*$
2. Project feature matrix onto this subspace, and estimate using robust principle component regression

Main Theorems

Conditions for recoverable subspace:

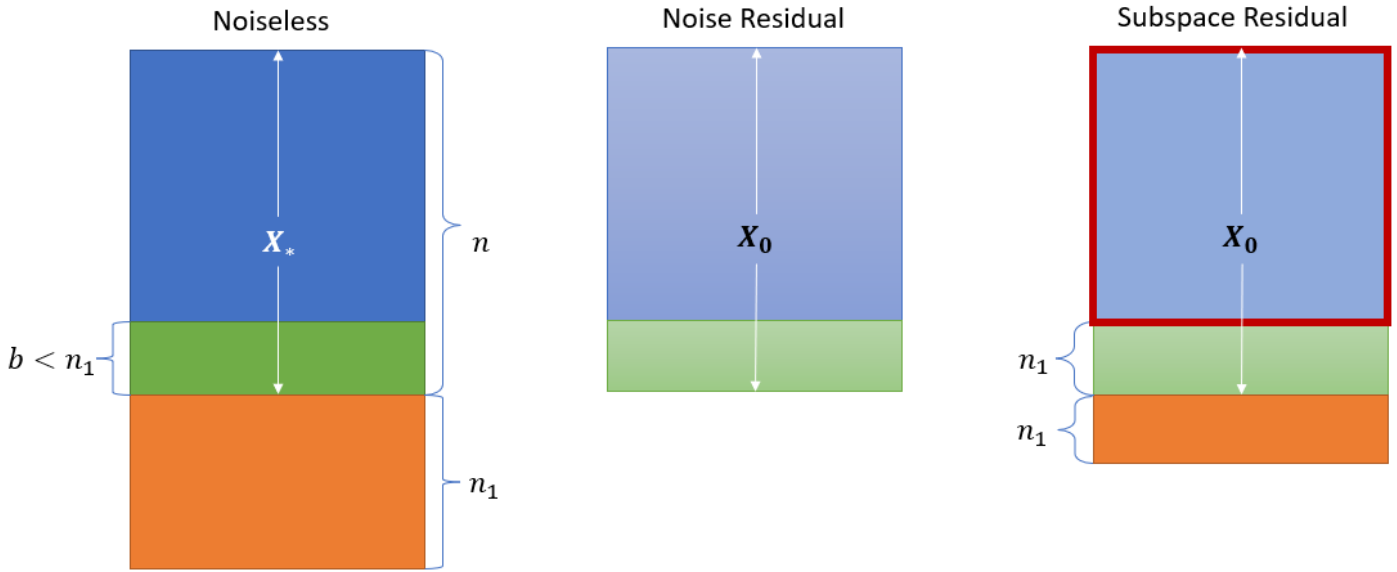
- **Theorems 1, 2:** If we do not have too many corrupt samples, we can recover the subspace *in the noiseless case*: $n_1 + MX_{k-1}(\boldsymbol{X}_*) < n$
- **Theorems 3, 4:** If the feature matrix is not so noisy that we can be fooled into thinking we're closer to a different adversarially-targeted basis, we can recover the subspace *in the noisy case.*

Algorithm for recovering the low-dimensional basis:

- **Theorem 5:** Trimmed Principal Component Regression recovers $\hat{\beta}$ such that the learner is tolerant to the attacks defined.

Step 1: Robust Subspace Recovery

How can an adversary influence subspace recovery (subspace of rank k?)



Step 2: Trimmed Principal Component Regression

Assume B is an orthogonal basis of k row vectors, and that $\mathbf{X}_* = \mathbf{U}_*\mathbf{B}$

We change the problem to $\mathbf{y}_* = \mathbf{X}_*\beta^* = \mathbf{U}\beta^*_U,$ where the adversary is allowed to corrupt U.

- Trim out the samples with the biggest $y_i - u_i\beta_U$
- We expect that the random noise on the label is small!

Algorithms

- Alternating minimization to find the best rank-k representation of X
- Trimmed optimization problems: also alternating minimization techniques

Trimmed optimization is not guaranteed to get global optima, but does well with random start.

Experimental results

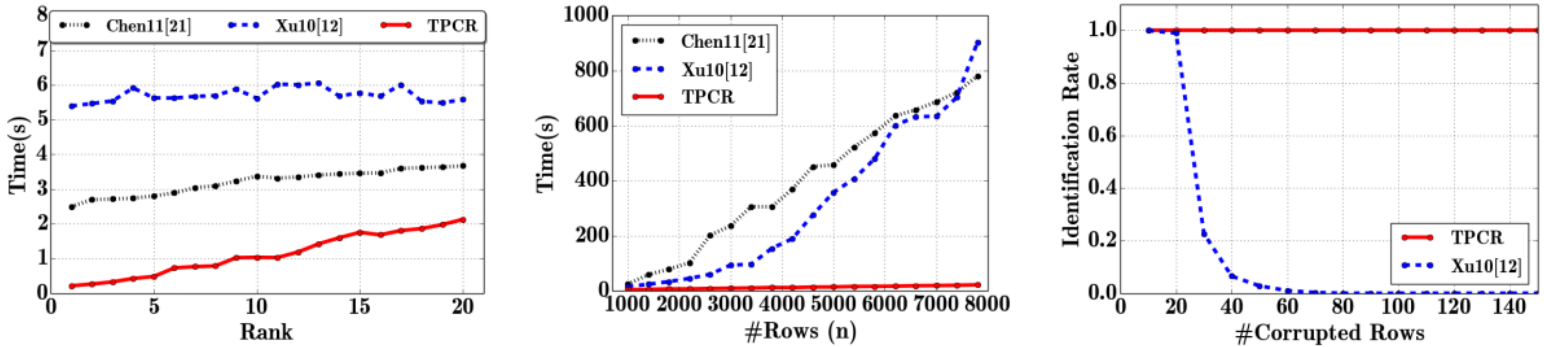
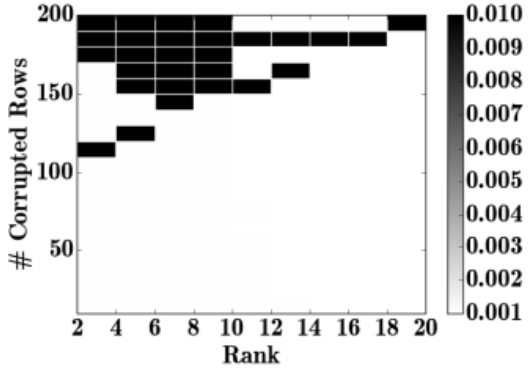
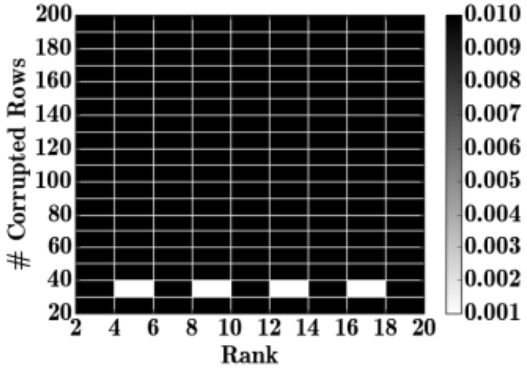


Figure 1: (a) Runtime, as a function of rank. (b) Runtime, as a function of the number of rows (n). (c) Rate of correct identification of corrupted rows.

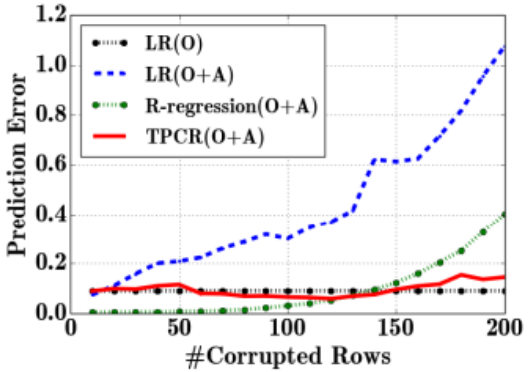
Experimental results



(a) TPCR



(b) Xu et al. [12]



(c)

Discussion

Pros:

- Robust to a constant number of corruptions, that can be corrupted in any manner, but can be maliciously targeting the subspace.
- Handles both noise, and deliberate corruption to the underlying subspace
- Efficiency/scalability

Cons

- Limited to linear regression
- Requires knowledge of the number of corrupted rows