Robust Statistics
and
Generative Adversarial Networks

Yuan YAO
HKUST
Deep Learning is Notoriously Not Robust!

- Imperceivable adversarial examples are ubiquitous to fail neural networks.
- How can one achieve robustness?

Outline
- Generalization and Breiman's Dilemma
- Robustness and Huber's Contamination Model
- Summary

Adversarial and Huber's Agnostic Contamination Model

- Deep Neural Networks are Notoriously not Robust

• Imperceivable adversarial examples are ubiquitous to fail neural networks

• How can one achieve robustness?
Robust Optimization

- Traditional training:
  \[
  \min_{\theta} J_n(\theta, z = (x_i, y_i)_{i=1}^n)
  \]
  - e.g. square or cross-entropy loss as negative log-likelihood of logit models

- Robust optimization (Madry et al. ICLR’2018):
  \[
  \min_{\theta} \max_{\|\epsilon_i\| \leq \delta} J_n(\theta, z = (x_i + \epsilon_i, y_i)_{i=1}^n)
  \]
  - robust to any distributions, yet computationally hard
Distributionally Robust Optimization (DRO)

- Distributional Robust Optimization:
  \[
  \min_{\theta} \max_{\epsilon} \mathbb{E}_{z \sim P_{\epsilon} \in \mathcal{D}} [J_n(\theta, z)]
  \]

- \(\mathcal{D}\) is a set of ambiguous distributions, e.g. Wasserstein ambiguity set
  \[
  \mathcal{D} = \{ P_{\epsilon} : W_2(P_{\epsilon}, \text{uniform distribution}) \leq \epsilon \}
  \]

where DRO may be reduced to regularized maximum likelihood estimates (Shafieezadeh-Abadeh, Esfahani, Kuhn, NIPS’2015) that are convex optimizations and tractable
Suppose that
\[ c \left( (x, y), (x', y') \right) = \begin{cases} \| x - x' \|_q^2 & \text{if } y = y' \\ \infty & \text{if } y \neq y' \end{cases}. \]

Then, if \( \frac{1}{p} + \frac{1}{q} = 1 \)

\[ \max_{P : D_c(P, P_n) \leq \delta} E_P^{1/2} \left( \left( Y - \beta^T X \right)^2 \right) = E_{P_n}^{1/2} \left( \left( Y - \beta^T X \right)^2 \right) + \sqrt{\delta} \| \beta \|_p. \]

**Remark 1:** This is sqrt-Lasso (Belloni et al. (2011)).

**Remark 2:** Uses RoPA duality theorem & "judicious choice of \( c (\cdot) \)"
Certified Robustness of Lasso

Take \( q = \infty \) and \( p = 1 \), with

\[
c \left( ((x, y), (x', y')) \right) = \begin{cases} 
\|x - x'\|^2_{\infty} & \text{if } y = y' \\
\infty & \text{if } y \neq y'
\end{cases}
\]

Then for

\[
P'_n = \frac{1}{n} \sum \delta_{x'_i}
\]

with \( \|x_i - x'_i\|_{\infty} \leq \delta \),

\[
D_c(P'_n, P_n) = \int \pi( ((x, y), (x', y')) ) c \left( ((x, y), (x', y')) \right) \leq \delta,
\]

for small enough \( \delta \) and well-separated \( x \)'s. Sqrt-Lasso

\[
\begin{aligned}
\min_{\beta} & \left\{ E_{P_n}^{1/2} \left[ \left( Y - \beta^T X \right)^2 \right] + \sqrt{\delta} \|\beta\|_1 \right\}^2 \\
= & \min_{\beta} \max_{P : D_c(P, P_n) \leq \delta} E_{P} \left( \left( Y - \beta^T X \right)^2 \right)
\end{aligned}
\]

provides a certified robust estimate in terms of Madry's adversarial training, using a convex Wasserstein relaxation.
TV-neighborhood

• Now how about the TV-uncertainty set?

$$\mathcal{D} = \{P_\varepsilon : TV(P_\varepsilon, \text{uniform distribution}) \leq \varepsilon\}?$$
Huber’s Model

\[ X_1, \ldots, X_n \sim (1 - \epsilon)P_\theta + \epsilon Q \]
Huber’s Model

\[ X_1, \ldots, X_n \sim (1 - \epsilon) P_{\theta} + \epsilon Q \]

parameter of interest

[Huber 1964]
Huber’s Model

\[ X_1, \ldots, X_n \sim (1 - \epsilon) P_\theta + \epsilon Q \]

contamination proportion

parameter of interest

[Huber 1964]
Huber’s Model

\[ X_1, ..., X_n \sim (1 - \varepsilon) P_\theta + \varepsilon Q \]

- contamination proportion
- arbitrary contamination
- parameter of interest

[Huber 1964]
An Example

\[ X_1, \ldots, X_n \sim (1 - \epsilon)N(\theta, I_p) + \epsilon Q. \]
An Example

\[ X_1, \ldots, X_n \sim (1 - \epsilon)N(\theta, I_p) + \epsilon Q. \]
Robust Maxmum-Likelihood
Does not work!

\[ X_1, \ldots, X_n \sim (1 - \epsilon)N(\theta, I_p) + \epsilon Q. \]
Robust Maximum-Likelihood
Does not work!

\[ X_1, \ldots, X_n \sim (1 - \epsilon)N(\theta, I_p) + \epsilon Q. \]

\[ \ell(\theta, Q) = \text{negative log-likelihood} = \sum_{i=1}^{n} (\theta - X_i)^2 \]

\[ \sim (1 - \epsilon)\mathbb{E}_{N(\theta)}(\theta - X)^2 + \epsilon \mathbb{E}_Q(\theta - X)^2 \]

the sample mean

\[ \hat{\theta}_{mean} = \frac{1}{n} \sum_{i=1}^{n} X_i = \arg \min_{\theta} \ell(\theta, Q) \]

\[ \min_{\theta} \max_{Q} \ell(\theta, Q) \geq \max_{Q} \min_{\theta} \ell(\theta, Q) = \max_{Q} \ell(\hat{\theta}_{mean}, Q) = \infty \]
Medians

1. Coordinatewise median

\[ \hat{\theta} = (\hat{\theta}_j), \text{ where } \hat{\theta}_j = \text{Median}\left(\{X_{ij}\}_{i=1}^n\right); \]
Medians

1. Coordinatewise median

\[ \hat{\theta} = (\hat{\theta}_j), \text{ where } \hat{\theta}_j = \text{Median}(\{X_{ij}\}_{i=1}^n); \]

2. Tukey’s median

\[ \hat{\theta} = \arg \max_{\eta \in \mathbb{R}^p} \min_{||u||=1} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i > u^T \eta\}. \]
## Comparisons

<table>
<thead>
<tr>
<th></th>
<th>Coordinatewise Median</th>
<th>Tukey’s Median</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>breakdown point</strong></td>
<td>$1/2$</td>
<td>$1/3$</td>
</tr>
<tr>
<td><strong>statistical precision</strong></td>
<td>$\frac{p}{n}$</td>
<td>$\frac{p}{n}$</td>
</tr>
<tr>
<td>(no contamination)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>statistical precision</strong></td>
<td>$\frac{p}{n} + p\epsilon^2$</td>
<td>$\frac{p}{n} + \epsilon^2$: minimax</td>
</tr>
<tr>
<td>(with contamination)</td>
<td></td>
<td>[Chen-Gao-Ren’15]</td>
</tr>
<tr>
<td><strong>computational complexity</strong></td>
<td>Polynomial</td>
<td>NP-hard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[Amenta et al. ’00]</td>
</tr>
</tbody>
</table>

Note: R-package for Tukey median can not deal with more than 10 dimensions!

[https://github.com/ChenMengjie/DepthDescent]
Multivariate Location Depth

\[
\left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i > u^T \eta\} \wedge \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i \leq u^T \eta\} \right\}
\]
\[
\min_{\|u\|=1} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i > u^T \eta\} \land \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i \leq u^T \eta\} \right\}
\]
Multivariate Location Depth

\[ \hat{\theta} = \arg \max_{\eta \in \mathbb{R}^p} \min_{\|u\| = 1} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i > u^T \eta\} \land \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i \leq u^T \eta\} \right\} \]
Multivariate Location Depth

$$\hat{\theta} = \arg \max_{\eta \in \mathbb{R}^p} \min_{\|u\| = 1} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i > u^T \eta\} \wedge \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i \leq u^T \eta\} \right\}$$

$$= \arg \max_{\eta \in \mathbb{R}^p} \min_{\|u\| = 1} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i > u^T \eta\}.$$
Regression Depth

model

\[ y|X \sim N(X^T \beta, \sigma^2) \]
Regression Depth

model \quad y|X \sim N(X^T \beta, \sigma^2)

embedding \quad Xy|X \sim N(XX^T \beta, \sigma^2 XX^T)
Regression Depth

model: \[ y | X \sim N(X^T \beta, \sigma^2) \]

embedding: \[ Xy | X \sim N(XX^T \beta, \sigma^2 XX^T) \]

projection: \[ u^T Xy | X \sim N(u^T XX^T \beta, \sigma^2 u^T XX^T u) \]
Regression Depth

model

\[ y|X \sim N(X^T \beta, \sigma^2) \]

embedding

\[ Xy|X \sim N(XX^T \beta, \sigma^2 XX^T) \]

projection

\[ u^T Xy|X \sim N(u^T XX^T \beta, \sigma^2 u^T XX^T u) \]

\[
\left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i (y_i - X_i^T \eta) > 0\} \wedge \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i (y_i - X_i^T \eta) \leq 0\} \right\}
\]
Regression Depth

model
\[ y|X \sim N(X^T \beta, \sigma^2) \]

embedding
\[ Xy|X \sim N(X X^T \beta, \sigma^2 X X^T) \]

projection
\[ u^T Xy|X \sim N(u^T X X^T \beta, \sigma^2 u^T X X^T u) \]

\[
\min_{u \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i (y_i - X_i^T \eta) > 0\} \wedge \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i (y_i - X_i^T \eta) \leq 0\} \right\}
\]
Regression Depth

model \( y|X \sim N(X^T \beta, \sigma^2) \)

embedding \( X y|X \sim N(X X^T \beta, \sigma^2 X X^T) \)

projection \( u^T X y|X \sim N(u^T X X^T \beta, \sigma^2 u^T X X^T u) \)

\[ \hat{\beta} = \arg\max_{\eta \in \mathbb{R}^p} \min_{u \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i (y_i - X_i^T \eta) > 0\} \land \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i (y_i - X_i^T \eta) \leq 0\} \right\} \]
Regression Depth

model

\[ y|X \sim N(X^T \beta, \sigma^2) \]

embedding

\[ Xy|X \sim N(XX^T \beta, \sigma^2 XX^T) \]

projection

\[ u^T Xy|X \sim N(u^T XX^T \beta, \sigma^2 u^T XX^T u) \]

\[
\hat{\beta} = \arg\max_{\eta \in \mathbb{R}^p} \min_{u \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i (y_i - X_i^T \eta) > 0\} \wedge \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i (y_i - X_i^T \eta) \leq 0\} \right\}
\]

[Rousseeuw & Hubert, 1999]
Tukey’s depth is not a special case of regression depth.
Multi-task Regression Depth

\[(X, Y) \in \mathbb{R}^p \times \mathbb{R}^m \sim \mathbb{P}\]
Multi-task Regression Depth

\[(X, Y) \in \mathbb{R}^p \times \mathbb{R}^m \sim \mathbb{P} \]

\[B \in \mathbb{R}^{p \times m}\]
Multi-task Regression Depth

\[(X, Y) \in \mathbb{R}^p \times \mathbb{R}^m \sim \mathbb{P}\]

\[B \in \mathbb{R}^{p \times m}\]

population version:

\[D_U(B, \mathbb{P}) = \inf_{U \in \mathcal{U}} \mathbb{P}\{\langle U^T X, Y - B^T X \rangle \geq 0\}\]

University of Chicago

December 19, 2016

Chao Gao
Multi-task Regression Depth

\[(X, Y) \in \mathbb{R}^p \times \mathbb{R}^m \sim \mathbb{P}\]

\[B \in \mathbb{R}^{p \times m}\]

population version:

\[D_U(B, \mathbb{P}) = \inf_{U \in \mathcal{U}} \mathbb{P} \{ \langle U^T X, Y - B^T X \rangle \geq 0 \}\]

empirical version:

\[D_U(B, \{(X_i, Y_i)\}_{i=1}^n) = \inf_{U \in \mathcal{U}} \frac{1}{n} \sum_{i=1}^n \mathbb{I} \{ \langle U^T X_i, Y_i - B^T X_i \rangle \geq 0 \}\]
Multi-task Regression Depth

$$(X, Y) \in \mathbb{R}^p \times \mathbb{R}^m \sim \mathbb{P}$$

$$B \in \mathbb{R}^{p \times m}$$

population version:

$$\mathcal{D}_U(B, \mathbb{P}) = \inf_{U \in \mathcal{U}} \mathbb{P} \left\{ \langle U^T X, Y - B^T X \rangle \geq 0 \right\}$$

empirical version:

$$\mathcal{D}_U(B, \{(X_i, Y_i)\}_{i=1}^n) = \inf_{U \in \mathcal{U}} \frac{1}{n} \sum_{i=1}^n \mathbb{I} \left\{ \langle U^T X_i, Y_i - B^T X_i \rangle \geq 0 \right\}$$

[Mizera, 2002]
Multi-task Regression Depth

$$D_U(B, \mathbb{P}) = \inf_{U \in \mathcal{U}} \mathbb{P} \{ \langle U^T X, Y - B^T X \rangle \geq 0 \}$$
Multi-task Regression Depth

\[ \mathcal{D}_U(B, \mathbb{P}) = \inf_{U \in \mathcal{U}} \mathbb{P} \{ \langle U^T X, Y - B^T X \rangle \geq 0 \} \]

\[ p = 1, \quad X = 1 \in \mathbb{R}, \]

\[ \mathcal{D}_U(b, \mathbb{P}) = \inf_{u \in \mathcal{U}} \mathbb{P} \{ u^T (Y - b) \geq 0 \} \]
Multi-task Regression Depth

\[ \mathcal{D}_U(B, \mathbb{P}) = \inf_{U \in \mathcal{U}} \mathbb{P} \{ \langle U^T X, Y - B^T X \rangle \geq 0 \} \]

\[ p = 1, X = 1 \in \mathbb{R}, \]

\[ \mathcal{D}_U(b, \mathbb{P}) = \inf_{u \in \mathcal{U}} \mathbb{P} \{ u^T (Y - b) \geq 0 \} \]

\[ m = 1, \]

\[ \mathcal{D}_U(\beta, \mathbb{P}) = \inf_{U \in \mathcal{U}} \mathbb{P} \{ u^T X (y - \beta^T X) \geq 0 \} \]
Proposition. For any $\delta > 0$,

$$
\sup_{B \in \mathbb{R}^{p \times m}} |\mathcal{D}(B, \mathbb{P}_n) - \mathcal{D}(B, \mathbb{P})| \leq C \sqrt{\frac{pm}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}},
$$

with probability at least $1 - 2\delta$. 

Multi-task Regression Depth

Theorem 4.1.

For any $\mathcal{N}$,

$$
\text{we have for any } \mathcal{N},
$$

Proposition 3.3.

Define $A, B, Q, C, C'$,

$$
\text{with } P, \mathbb{P}, \mathcal{D}(B, \mathbb{P}), \mathcal{D}(B, \mathbb{P}_n).
$$

Proposition 4.1.

$$
\text{For any probability measure } \mathbb{P}.
\text{We have for any } \mathcal{N},
$$

Theorem 3.4.

$$
\text{Proposition 3.3}.
\text{We have for any } \mathcal{N}.
$$

4.1 Multiple Linear Regression

4 Applications of Multi-task Regression Depth

\text{Define } A, B, Q, C, C'.

\text{Proposition 3.3.}

$$
\text{For any probability measure } \mathbb{P}.
\text{We have for any } \mathcal{N},
$$

Theorem 4.1.

$$
\text{Proposition 3.3}.
\text{We have for any } \mathcal{N}.
$$

\text{4.1 Multiple Linear Regression}

\text{4 Applications of Multi-task Regression Depth}

\text{Proposition 3.3.}

$$
\text{For any probability measure } \mathbb{P}.
\text{We have for any } \mathcal{N},
$$

Theorem 4.1.

$$
\text{Proposition 3.3}.
\text{We have for any } \mathcal{N}.
$$

\text{4.1 Multiple Linear Regression}

\text{4 Applications of Multi-task Regression Depth}

\text{Proposition 3.3.}

$$
\text{For any probability measure } \mathbb{P}.
\text{We have for any } \mathcal{N},
$$

Theorem 4.1.

$$
\text{Proposition 3.3}.
\text{We have for any } \mathcal{N}.
$$
Multi-task Regression Depth

**Proposition.** For any $\delta > 0$, 
\[
\sup_{B \in \mathbb{R}^{p \times m}} |\mathcal{D}(B, \mathbb{P}_n) - \mathcal{D}(B, \mathbb{P})| \leq C \sqrt{\frac{pm}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}}, 
\]
with probability at least $1 - 2\delta$.

**Proposition.** 
\[
\sup_{B, Q} |\mathcal{D}(B, (1 - \epsilon P_{B^*}) + \epsilon Q) - \mathcal{D}(B, P_{B^*})| \leq \epsilon
\]
Multi-task Regression Depth

\[ (X, Y) \sim P_B \]
Multi-task Regression Depth

\[(X, Y) \sim P_B : X \sim N(0, \Sigma), \quad Y|X \sim N(B^T X, \sigma^2 I_m)\]
Multi-task Regression Depth

\[(X, Y) \sim P_B : X \sim N(0, \Sigma), \quad Y|X \sim N(B^T X, \sigma^2 I_m)\]

\[(X_1, Y_1), \ldots, (X_n, Y_n) \sim (1 - \epsilon)P_B + \epsilon Q\]
Multi-task Regression Depth

\[(X, Y) \sim P_B : X \sim N(0, \Sigma), \quad Y | X \sim N(B^TX, \sigma^2 I_m)\]

\[(X_1, Y_1), \ldots, (X_n, Y_n) \sim (1 - \epsilon)P_B + \epsilon Q\]

**Theorem [G17].** For some \(C > 0\),

\[
\text{Tr}((\hat{B} - B)^T \Sigma (\hat{B} - B)) \leq C\sigma^2 \left( \frac{pm}{n} \vee \epsilon^2 \right),
\]

\[
\| \hat{B} - B \|^2_F \leq C\frac{\sigma^2}{\kappa^2} \left( \frac{pm}{n} \vee \epsilon^2 \right),
\]

with high probability uniformly over \(B, Q\).
Covariance Matrix

\[ X_1, \ldots, X_n \sim (1 - \epsilon)N(0, \Sigma) + \epsilon Q. \]
Covariance Matrix

\[ X_1, \ldots, X_n \sim (1 - \epsilon)N(0, \Sigma) + \epsilon Q. \]

How to estimate \( \Sigma \)?
Covariance Matrix
Covariance Matrix
Covariance Matrix
Covariance Matrix
Covariance Matrix
Covariance Matrix

\[ \mathcal{D}(\Gamma, \{X_i\}_{i=1}^n) = \min_{\|u\|=1} \min \left\{ \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{|u^T X_i|^2 \geq u^T \Gamma u\}, \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{|u^T X_i|^2 < u^T \Gamma u\} \right\} \]
Covariance Matrix

\[ \mathcal{D}(\Gamma, \{X_i\}_{i=1}^n) = \min_{\|u\|=1} \min \left\{ \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{|u^T X_i|^2 \geq u^T \Gamma u\}, \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{|u^T X_i|^2 < u^T \Gamma u\} \right\} \]

\[ \hat{\Gamma} = \arg \max_{\Gamma \geq 0} \mathcal{D}(\Gamma, \{X_i\}_{i=1}^n) \quad \hat{\Sigma} = \hat{\Gamma} / \beta \]
Covariance Matrix

\[ \mathcal{D}(\Gamma, \{X_i\}_{i=1}^n) = \min_{\|u\|=1} \min \left\{ \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{|u^T X_i|^2 \geq u^T \Gamma u\}, \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{|u^T X_i|^2 < u^T \Gamma u\} \right\} \]

\[ \hat{\Gamma} = \arg \max_{\Gamma \geq 0} \mathcal{D}(\Gamma, \{X_i\}_{i=1}^n) \quad \hat{\Sigma} = \hat{\Gamma} / \beta \]

**Theorem [CGR15].** For some \( C > 0 \),

\[ \|\hat{\Sigma} - \Sigma\|_\text{op}^2 \leq C \left( \frac{p}{n} \vee \epsilon^2 \right) \]

with high probability uniformly over \( \Sigma, Q \).
<table>
<thead>
<tr>
<th>Model</th>
<th>Norm</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>$| \cdot |^2$</td>
<td>$\frac{p}{n} \sqrt{\epsilon^2}$</td>
</tr>
<tr>
<td>reduced rank regression</td>
<td>$| \cdot |_F^2$</td>
<td>$\frac{\sigma^2}{\kappa^2} \frac{r(p+m)}{n} \sqrt{\frac{\sigma^2}{\kappa^2} \epsilon^2}$</td>
</tr>
<tr>
<td>Gaussian graphical model</td>
<td>$| \cdot |_{\ell_1}^2$</td>
<td>$\frac{s^2 \log(ep/s)}{n} \sqrt{s \epsilon^2}$</td>
</tr>
<tr>
<td>covariance matrix</td>
<td>$| \cdot |_{op}^2$</td>
<td>$\frac{p}{n} \sqrt{\epsilon^2}$</td>
</tr>
<tr>
<td>sparse PCA</td>
<td>$| \cdot |_F^2$</td>
<td>$\frac{s \log(ep/s)}{n \lambda^2} \sqrt{\frac{\epsilon^2}{\lambda^2}}$</td>
</tr>
</tbody>
</table>
## Summary

<table>
<thead>
<tr>
<th>Model</th>
<th>Norm</th>
<th>Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>| \cdot | ^2</td>
<td>\frac{p}{n} \sqrt{\epsilon^2}</td>
</tr>
<tr>
<td>reduced rank regression</td>
<td>| \cdot | _F ^2</td>
<td>\frac{\sigma^2 r(p+m)}{\kappa^2 n} \sqrt{\frac{\sigma^2}{\kappa^2 \epsilon^2}}</td>
</tr>
<tr>
<td>Gaussian graphical model</td>
<td>| \cdot | _{\ell_1} ^2</td>
<td>\frac{s^2 \log(ep/s)}{n} \sqrt{s \epsilon^2}</td>
</tr>
<tr>
<td>covariance matrix</td>
<td>| \cdot | _{op} ^2</td>
<td>\frac{p}{n} \sqrt{\epsilon^2}</td>
</tr>
<tr>
<td>sparse PCA</td>
<td>| \cdot | _F ^2</td>
<td>\frac{s \log(ep/s)}{n \lambda^2} \sqrt{\frac{\epsilon^2}{\lambda^2}}</td>
</tr>
</tbody>
</table>
Computation
Computational Challenges

\[ X_1, \ldots, X_n \sim (1 - \epsilon)N(\theta, I_p) + \epsilon Q. \]
How to estimate $\theta$?

**Computational Challenges**

$Lai,~Rao,~Vempala$

$Diakonikolas,~Kamath,~Kane,~Li,~Moitra,~Stewart$

$Balakrishnan,~Du,~Singh$

\[ X_1, \ldots, X_n \sim (1 - \epsilon)N(\theta, I_p) + \epsilon Q. \]
Computational Challenges

\[ X_1, \ldots, X_n \sim (1 - \epsilon)N(\theta, I_p) + \epsilon Q. \]

Lai, Rao, Vempala
Diakonikolas, Kamath, Kane, Li, Moitra, Stewart
Balakrishnan, Du, Singh

- Polynomial algorithms are proposed [Diakonikolas et al.'16, Lai et al. 16]
  of minimax optimal statistical precision
  - needs information on second or higher order of moments
  - some priori knowledge about \( \epsilon \)
Advantages of Tukey Median

• A well-defined objective function
• Adaptive to and
• Optimal for any elliptical distribution
Advantages of Tukey Median

- A well-defined objective function

Chao Gao, Department of Statistics, Yale University  

August 12, 2018
Advantages of Tukey Median

• A well-defined objective function
• Adaptive to \(\epsilon\) and \(\sum\)
Advantages of Tukey Median

- A well-defined objective function
- Adaptive to $\epsilon$ and $\sum$
- Optimal for any elliptical distribution
A practically good algorithm?
Generative Adversarial Networks [Goodfellow et al. 2014]

Note: R-package for Tukey median can not deal with more than 10 dimensions [https://github.com/ChenMengjie/DepthDescent]
Robust Learning of Cauchy Distributions

Table 4: Comparison of various methods of robust location estimation under Cauchy distributions. Samples are drawn from \((1 - \epsilon)\text{Cauchy}(0_p, I_p) + \epsilon Q\) with \(\epsilon = 0.2, p = 50\) and various choices of \(Q\). Sample size: 50,000. Discriminator net structure: 50-50-25-1. Generator \(g_\omega(\xi)\) structure: 48-48-32-24-12-1 with absolute value activation function in the output layer.

<table>
<thead>
<tr>
<th>Contamination (Q)</th>
<th>JS-GAN ((G_1))</th>
<th>JS-GAN ((G_2))</th>
<th>Dimension Halving</th>
<th>Iterative Filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cauchy((1.5 \times 1_p, I_p))</td>
<td>0.0664 (0.0065)</td>
<td>0.0743 (0.0103)</td>
<td>0.3529 (0.0543)</td>
<td>0.1244 (0.0114)</td>
</tr>
<tr>
<td>Cauchy((5.0 \times 1_p, I_p))</td>
<td>0.0480 (0.0058)</td>
<td>0.0540 (0.0064)</td>
<td>0.4855 (0.0616)</td>
<td>0.1687 (0.0310)</td>
</tr>
<tr>
<td>Cauchy((1.5 \times 1_p, 5 \times I_p))</td>
<td>0.0754 (0.0135)</td>
<td>0.0742 (0.0111)</td>
<td>0.3726 (0.0530)</td>
<td>0.1220 (0.0112)</td>
</tr>
<tr>
<td>Normal((1.5 \times 1_p, 5 \times I_p))</td>
<td>0.0702 (0.0064)</td>
<td>0.0713 (0.0088)</td>
<td>0.3915 (0.0232)</td>
<td>0.1048 (0.0288)</td>
</tr>
</tbody>
</table>

- **Dimension Halving:** [Lai et al.’16]

- **Iterative Filtering:** [Diakonikolas et al.’17]
**f-GAN**

Given a strictly convex function $f$ that satisfies $f(1) = 0$, the $f$-divergence between two probability distributions $P$ and $Q$ is defined by

$$D_f(P \| Q) = \int f\left(\frac{p}{q}\right) dQ.$$  

Let $f^*$ be the convex conjugate of $f$. A variational lower bound of (8) is

$$D_f(P \| Q) \geq \sup_{T \in \mathcal{T}} \left[ \mathbb{E}_P T(X) - \mathbb{E}_Q f^*(T(X)) \right].$$  

where equality holds whenever the class $\mathcal{T}$ contains the function $f'(p/q)$.

[Nowozin-Cseke-Tomioka'16] $f$-GAN minimizes the variational lower bound (9)

$$\hat{P} = \arg \min_{Q \in \mathcal{Q}} \sup_{T \in \mathcal{T}} \left[ \frac{1}{n} \sum_{i=1}^{n} T(X_i) - \mathbb{E}_Q f^*(T(X)) \right].$$

with i.i.d. observations $X_1, \ldots, X_n \sim P$. 

From f-GAN to Tukey’s Median: f-learning

Consider the special case

\[ \mathcal{T} = \left\{ f' \left( \frac{\tilde{q}}{q} \right) : \tilde{q} \in \tilde{Q} \right\}. \]  

which is tight if \( P \in \tilde{Q} \). The sample version leads to the following f-learning

\[ \hat{P} = \arg \min_{Q \in \mathcal{Q}} \sup_{\tilde{Q} \in \tilde{Q}} \left[ \frac{1}{n} \sum_{i=1}^{n} f' \left( \frac{\tilde{q}(X_i)}{q(X_i)} \right) - \mathbb{E}_Q f^* \left( f' \left( \frac{\tilde{q}(X)}{q(X)} \right) \right) \right]. \]  

• If \( f(x) = x \log x \), \( Q = \tilde{Q} \), (12) ⇒ Maximum Likelihood Estimate

• If \( f(x) = (x - 1)^+ \), then \( D_f (P \| Q) = \frac{1}{2} \int |p - q| \) is the TV-distance, \( f^*(t) = t \mathbb{I} \{ 0 \leq t \leq 1 \} \), f-GAN ⇒ TV-GAN

• \( Q = \{ N(\eta, l_p) : \eta \in \mathbb{R}^p \} \) and \( \tilde{Q} = \{ N(\tilde{\eta}, l_p) : \|\tilde{\eta} - \eta\| \leq r \} \), (12) \( r \rightarrow 0 \)

Tukey’s Median
f-Learning
An f-divergence is defined as

$$D_f(P\|Q) = \int f \left( \frac{p}{q} \right) dQ$$

Since $f(u) = \sup_t (tu f^\star(t))$, it is not hard to derive the following variational form of f-divergence,

$$D_f(P\|Q) = \sup_T \left[ E_X \sim P T(X) - E_X \sim Q f^\star(T(X)) \right].$$

The optimal $T$ is achieved by $T(x) = f_0 \left( \frac{p(x)}{q(x)} \right)$.

GAN is a special case of f-GAN by taking $f(x) = x \log x - \log(x + 1) \log(x + 1)$.

Its conjugate function is $f^\star(t) = \log(e^t - 1)$.

Therefore, with this particular $f$, we get

$$D_f(P\|Q) = \sup_T \left[ E_X \sim P T(X) + E_X \sim Q \log(e^T(X) - 1) \right].$$

With the transformation $T(x) = \log D(x)$, we recover the original definition of GAN.

Similar to GAN, we can consider a symmetric class of $T$. This leads to the estimation procedure

$$\min_Q \max_{\tilde{Q}} \sum \log \tilde{q}(x_i) q(x_i) = \int f_0 \left( \frac{\tilde{q}}{q} \right) dQ,$$

which is the general density estimation procedure of f-GAN.

A special choice of $f$ is $f(x) = x \log x$, which leads to the KL-divergence $D_f(P\|Q) = D_{KL}(P\|Q)$. For this $f$, its derivative and conjugate functions are

$$f_0(x) = 1 + \log x,$$

and

$$f^\star(t) = e^{t} - 1.$$

Then, the procedure becomes

$$\min_Q \max_{\tilde{Q}} \sum \log \tilde{q}(x_i) q(x_i) = 2 \min_Q \max_{\tilde{Q}} \sum \log \frac{q(x_i)}{\tilde{q}(x_i)},$$

which is the MLE.
f-Learning

f-divergence

\[ D_f(P \| Q) = \int f \left( \frac{p}{q} \right) dQ \]

\[ f(u) = \sup_t (tu - f^*(t)) \]
\textbf{f-Learning}

\textbf{f-divergence} \quad \mathcal{D}_f(P \| Q) = \int f \left( \frac{p}{q} \right) dQ

\textbf{variational representation} \quad = \sup \limits_{T} \left[ \mathbb{E}_{X \sim P} T(X) - \mathbb{E}_{X \sim Q} f^*(T(X)) \right]
f-Learning

\[ D_f(P \| Q) = \int f \left( \frac{p}{q} \right) dQ \]

\[ = \sup_T \left[ \mathbb{E}_{X \sim P} T(X) - \mathbb{E}_{X \sim Q} f^*(T(X)) \right] \]

\[ T(x) = f' \left( \frac{p(x)}{q(x)} \right) \]
f-Learning

**f-divergence**

\[ D_f(P\|Q) = \int f \left( \frac{p}{q} \right) dQ \]

**variational representation**

\[ = \sup_{T} \left[ \mathbb{E}_{X \sim P} T(X) - \mathbb{E}_{X \sim Q} f^*(T(X)) \right] \]

\[ = \sup_{\tilde{Q}} \left\{ \mathbb{E}_{X \sim P} f' \left( \frac{d\tilde{Q}(X)}{dQ(X)} \right) - \mathbb{E}_{X \sim Q} f^* \left( f' \left( \frac{d\tilde{Q}(X)}{dQ(X)} \right) \right) \right\} \]
f-Learning

\[
\max_{T \in \mathcal{T}} \left\{ \frac{1}{n} \sum_{i=1}^{n} T(X_i) - \int f^* (T) \, dQ \right\}
\]

\[
\max_{\tilde{Q} \in \tilde{Q}} \left\{ \frac{1}{n} \sum_{i=1}^{n} f' \left( \frac{\tilde{q}(X_i)}{q(X_i)} \right) - \int f^* \left( f' \left( \frac{\tilde{q}}{q} \right) \right) \, dQ \right\}
\]
f-Learning

\[
\min_{Q \in \mathcal{Q}} \max_{T \in \mathcal{T}} \left\{ \frac{1}{n} \sum_{i=1}^{n} T(X_i) - \int f^* (T) \, dQ \right\}
\]

\[
\min_{Q \in \mathcal{Q}} \max_{\hat{Q} \in \hat{Q}} \left\{ \frac{1}{n} \sum_{i=1}^{n} f'(\frac{\tilde{q}(X_i)}{q(X_i)}) - \int f^* \left( f' \left( \frac{\tilde{q}}{q} \right) \right) \, dQ \right\}
\]
f-Learning

\[
\text{f-GAN} \quad \min_{Q \in \mathcal{Q}} \max_{T \in \mathcal{T}} \left\{ \frac{1}{n} \sum_{i=1}^{n} T(X_i) - \int f^* (T) \, dQ \right\}
\]

\[
\text{f-Learning} \quad \min_{Q \in \mathcal{Q}} \max_{\tilde{Q} \in \tilde{\mathcal{Q}}} \left\{ \frac{1}{n} \sum_{i=1}^{n} f'(\frac{\tilde{q}(X_i)}{q(X_i)}) - \int f^* \left( f' \left( \frac{\tilde{q}}{q} \right) \right) \, dQ \right\}
\]
f-Learning

f-GAN
\[
\min_{Q \in Q} \max_{T \in T} \left\{ \frac{1}{n} \sum_{i=1}^{n} T(X_i) - \int f^*(T) \, dQ \right\}
\]

f-Learning
\[
\min_{Q \in Q} \max_{\hat{Q} \in \hat{Q}} \left\{ \frac{1}{n} \sum_{i=1}^{n} f' \left( \frac{\hat{q}(X_i)}{q(X_i)} \right) - \int f^* \left( f' \left( \frac{\hat{q}}{q} \right) \right) \, dQ \right\}
\]

[Nowozin, Cseke, Tomioka]
f-Learning
## f-Learning

<table>
<thead>
<tr>
<th>Jensen-Shannon</th>
<th>( f(x) = x \log x - (x + 1) \log(x + 1) )</th>
<th>GAN</th>
</tr>
</thead>
</table>

[Goodfellow et al.]
## f-Learning

<table>
<thead>
<tr>
<th>Jensen-Shannon</th>
<th>$f(x) = x \log x - (x + 1) \log(x + 1)$</th>
<th>GAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kullback-Leibler</td>
<td>$f(x) = x \log x$</td>
<td>MLE</td>
</tr>
</tbody>
</table>

[Goodfellow et al.]

# f-Learning

<table>
<thead>
<tr>
<th></th>
<th>Function</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jensen-Shannon</td>
<td>$f(x) = x \log x - (x + 1) \log(x + 1)$</td>
<td>GAN</td>
</tr>
<tr>
<td>Kullback-Leibler</td>
<td>$f(x) = x \log x$</td>
<td>MLE</td>
</tr>
<tr>
<td>Hellinger Squared</td>
<td>$f(x) = 2 - 2\sqrt{x}$</td>
<td>$\rho$</td>
</tr>
</tbody>
</table>

[Goodfellow et al., Baraud and Birge]
f-Learning

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Jensen-Shannon</strong></td>
<td>$f(x) = x \log x - (x + 1) \log(x + 1)$</td>
<td>GAN</td>
</tr>
<tr>
<td><strong>Kullback-Leibler</strong></td>
<td>$f(x) = x \log x$</td>
<td>MLE</td>
</tr>
<tr>
<td><strong>Hellinger Squared</strong></td>
<td>$f(x) = 2 - 2\sqrt{x}$</td>
<td>rho</td>
</tr>
<tr>
<td><strong>Total Variation</strong></td>
<td>$f(x) = (x - 1)_+$</td>
<td>depth</td>
</tr>
</tbody>
</table>

[Goodfellow et al., Baraud and Birge]
TV-Learning

\[
\min_{Q \in \mathcal{Q}} \max_{\tilde{Q} \in \tilde{\mathcal{Q}}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left\{ \frac{\tilde{q}(X_i)}{q(X_i)} \geq 1 \right\} - Q \left( \frac{\tilde{q}}{q} \geq 1 \right) \right\}
\]
TV-Learning

\[
\min_{Q \in \mathcal{Q}} \max_{\tilde{Q} \in \tilde{\mathcal{Q}}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left\{ \frac{\tilde{q}(X_i)}{q(X_i)} \geq 1 \right\} - Q \left( \frac{\tilde{q}}{q} \geq 1 \right) \right\}
\]

\[
\mathcal{Q} = \left\{ N(\theta, I_p) : \theta \in \mathbb{R}^p \right\} \quad \tilde{\mathcal{Q}} = \left\{ N(\tilde{\theta}, I_p) : \tilde{\theta} \in \mathcal{N}_r(\theta) \right\}
\]
TV-Learning

$$\min_{Q \in \mathcal{Q}} \max_{\tilde{Q} \in \tilde{\mathcal{Q}}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left\{ \frac{\tilde{q}(X_i)}{q(X_i)} \geq 1 \right\} - Q \left( \frac{\tilde{q}}{q} \geq 1 \right) \right\}$$

\[ Q = \left\{ N(\theta, I_p) : \theta \in \mathbb{R}^p \right\} \quad \tilde{Q} = \left\{ N(\tilde{\theta}, I_p) : \tilde{\theta} \in \mathcal{N}_r(\theta) \right\} \]

$r \to 0$
TV-Learning

$$\min_{Q \in \mathcal{Q}} \max_{\tilde{Q} \in \tilde{\mathcal{Q}}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left\{ \frac{\tilde{q}(X_i)}{q(X_i)} \geq 1 \right\} - Q \left( \frac{\tilde{q}}{q} \geq 1 \right) \right\}$$

$$\mathcal{Q} = \left\{ N(\theta, I_p) : \theta \in \mathbb{R}^p \right\} \quad \tilde{\mathcal{Q}} = \left\{ N(\tilde{\theta}, I_p) : \tilde{\theta} \in \mathcal{N}_r(\theta) \right\}$$

$\overset{r \to 0}{\longrightarrow}$

Tukey depth

$$\max_{\theta \in \mathbb{R}} \min_{\|u\|=1} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left\{ \mathbf{u}^T \mathbf{X}_i \geq \mathbf{u}^T \mathbf{\theta} \right\}$$
TV-Learning

\[
\min_{Q \in \mathcal{Q}} \max_{\tilde{Q} \in \tilde{\mathcal{Q}}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\left\{ \frac{\hat{q}(X_i)}{q(X_i)} \geq 1 \right\} - Q\left( \frac{\hat{q}}{q} \geq 1 \right) \right\}
\]
TV-Learning

$$\min_{Q \in \mathcal{Q}} \max_{\tilde{Q} \in \tilde{\mathcal{Q}}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left\{ \frac{\tilde{q}(X_i)}{q(X_i)} \geq 1 \right\} - Q \left( \frac{\tilde{q}}{q} \geq 1 \right) \right\}$$

$$\mathcal{Q} = \left\{ N(0, \Sigma) : \Sigma \in \mathbb{R}^{p \times p} \right\} \quad \tilde{\mathcal{Q}} = \left\{ N(0, \tilde{\Sigma}) : \tilde{\Sigma} = \Sigma + ruu^T, \|u\| = 1 \right\}$$
TV-Learning

\[
\min_{Q \in \mathcal{Q}} \max_{\tilde{Q} \in \tilde{\mathcal{Q}}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left\{ \frac{\tilde{q}(X_i)}{q(X_i)} \geq 1 \right\} - Q \left( \frac{\tilde{q}}{q} \geq 1 \right) \right\}
\]

\[
\mathcal{Q} = \left\{ N(0, \Sigma) : \Sigma \in \mathbb{R}^{p \times p} \right\} \quad \tilde{\mathcal{Q}} = \left\{ N(0, \tilde{\Sigma}) : \tilde{\Sigma} = \Sigma + ruu^T, \|u\| = 1 \right\}
\]

\[
r \to 0
\]
TV-Learning

$$\min_{Q \in \mathcal{Q}} \max_{\tilde{Q} \in \tilde{\mathcal{Q}}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left\{ \frac{\tilde{q}(X_i)}{q(X_i)} \geq 1 \right\} - Q \left( \frac{\tilde{q}}{q} \geq 1 \right) \right\}$$

\[ \mathcal{Q} = \left\{ N(0, \Sigma) : \Sigma \in \mathbb{R}^{p \times p} \right\} \quad \tilde{\mathcal{Q}} = \left\{ N(0, \tilde{\Sigma}) : \tilde{\Sigma} = \Sigma + ruu^T, \|u\| = 1 \right\} \]

(related to) matrix depth

$$\max_{\Sigma} \min_{\|u\| = 1} \left[ \left( \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{|u^T X_i|^2 \leq u^T \Sigma u\} - \mathbb{P}(\chi_1^2 \leq 1) \right) \land \left( \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{|u^T X_i|^2 > u^T \Sigma u\} - \mathbb{P}(\chi_1^2 > 1) \right) \right]$$

$r \to 0$
robust statistics community  
deep learning community
robust statistics community

f-Learning f-GAN

deep learning community
robust statistics community

f-Learning
f-GAN

deep learning community

practically good algorithms
robust statistics community

f-Learning

f-GAN

theoretical foundation

deep learning community

practically good algorithms
TV-GAN

\[ \hat{\theta} = \arg\min_{\eta} \sup_{w,b} \left[ \frac{1}{n} \sum_{i=1}^{n} \frac{1}{1 + e^{-w^T X_i - b}} - E_{\eta} \frac{1}{1 + e^{-w^T X - b}} \right] \]
TV-GAN

$$\hat{\theta} = \arg\min_{\eta} \sup_{w,b} \left[ \frac{1}{n} \sum_{i=1}^{n} \frac{1}{1 + e^{-w^T X_i - b}} - E_{\eta} \frac{1}{1 + e^{-w^T X - b}} \right]$$

$$N(\eta, I_p)$$
TV-GAN

\[ \hat{\theta} = \arg\min_{\eta} \sup_{w,b} \left[ \frac{1}{n} \sum_{i=1}^{n} \frac{1}{1 + e^{-w^T X_i - b}} - E_{\eta} \frac{1}{1 + e^{-w^T X - b}} \right] \]

logistic regression classifier

\( N(\eta, I_p) \)
**TV-GAN**

\[
\hat{\theta} = \arg\min_{\eta} \sup_{w,b} \left[ \frac{1}{n} \sum_{i=1}^{n} \frac{1}{1 + e^{-w^T X_i - b}} - E_{\eta} \frac{1}{1 + e^{-w^T X - b}} \right]
\]

logistic regression classifier

**Theorem [GLYZ18].** For some \( C > 0, \)

\[
\left\| \hat{\theta} - \theta \right\|^2 \leq C \left( \frac{p}{n} \vee \epsilon^2 \right)
\]

with high probability uniformly over \( \theta \in \mathbb{R}^p, Q. \)
TV-GAN
rugged landscape!

**Figure:** Heatmaps of the landscape of $F(\eta, w) = \sup_b[^{EP}\text{sigmoid}(wX + b) - ^{EN(\eta, 1)}\text{sigmoid}(wX + b)]$, where $b$ is maximized out for visualization. Left: samples are drawn from $P = (1 - \epsilon)N(1, 1) + \epsilon N(1.5, 1)$ with $\epsilon = 0.2$. Right: samples are drawn from $P = (1 - \epsilon)N(1, 1) + \epsilon N(10, 1)$ with $\epsilon = 0.2$. Left: the landscape is good in the sense that no matter whether we start from the left-top area or the right-bottom area of the heatmap, gradient ascent on $\eta$ does not consistently increase or decrease the value of $\eta$. This is because the signal becomes weak when it is close to the saddle point around $\eta = 1$. Right: it is clear that $\tilde{F}(w) = F(\eta, w)$ has two local maxima for a given $\eta$, achieved at $w = +\infty$ and $w = -\infty$. In fact, the global maximum for $\tilde{F}(w)$ has a phase transition from $w = +\infty$ to $w = -\infty$ as $\eta$ grows. For example, the maximum is achieved at $w = +\infty$ when $\eta = 1$ (blue solid) and is achieved at $w = -\infty$ when $\eta = 5$ (red solid). Unfortunately, even if we initialize with $\eta_0 = 1$ and $w_0 > 0$, gradient ascents on $\eta$ will only increase the value of $\eta$ (green dash), and thus as long as the discriminator cannot reach the global maximizer, $w$ will be stuck in the positive half space $\{w : w > 0\}$ and further increase the value of $\eta$. 
The Original JS-GAN

[Goodfellow et al. 2014] For \( f(x) = x \log x - (x + 1) \log \frac{x+1}{2} \),

\[
\hat{\theta} = \arg \min_{\eta \in \mathbb{R}^p} \max_{D \in \mathcal{D}} \left[ \frac{1}{n} \sum_{i=1}^{n} \log D(X_i) + \mathbb{E}_{\mathcal{N}(\eta, I_p)} \log(1 - D(X)) \right] + \log 4. \tag{15}
\]

What are \( \mathcal{D} \), the class of discriminators?

- Single layer (no hidden layer):
  \[
  \mathcal{D} = \left\{ D(x) = \text{sigmoid}(w^T x + b) : w \in \mathbb{R}^p, b \in \mathbb{R} \right\}
  \]

- One-hidden or Multiple layer:
  \[
  \mathcal{D} = \left\{ D(x) = \text{sigmoid}(w^T g(X)) \right\}
  \]
JS-GAN

\[ \hat{\theta} = \arg\min_{\eta \in \mathbb{R}^p} \max_{T \in \mathcal{T}} \left[ \frac{1}{n} \sum_{i=1}^{n} \log T(X_i) + E_{\eta} \log(1 - T(X)) \right] + \log 4 \]
\[ \hat{\theta} = \arg\min_{\eta \in \mathbb{R}^p} \max_{T \in T} \left[ \frac{1}{n} \sum_{i=1}^{n} \log T(X_i) + E_{\eta} \log(1 - T(X)) \right] + \log 4 \]

**numerical experiment**

\[ X_1, ..., X_n \sim (1 - \epsilon)N(\theta, I_p) + \epsilon N(\tilde{\theta}, I_p) \]
The original proposal of GAN that an algorithm in \( \mathbb{R}^p \) takes results also include the behavior of error against dimension. When to making angle to study the robustness of various f-GAN procedures. Our research question is: Remarkably, it is robust because the relation (Theorem 21) can be derived from basic f-divergence inequalities. This means \( \text{argmax} \) \( \eta \in \mathbb{R}^p \) \( T \in \mathcal{T} \left[ \frac{1}{n} \sum_{i=1}^{n} \log T(X_i) + E_n \log(1 - T(X)) \right] \) + \log 4.

\[ \hat{\theta} = \arg\min_{\eta \in \mathbb{R}^p} \max_{T \in \mathcal{T}} \left[ \frac{1}{n} \sum_{i=1}^{n} \log T(X_i) + E_n \log(1 - T(X)) \right] + \log 4 \]

**numercial experiment**

\[ X_1, \ldots, X_n \sim (1 - \epsilon)N(\theta, I_p) + \epsilon N(\tilde{\theta}, I_p) \]
Moreover, we also know that Kullback-Leibler learning (MLE) is not robust, because

The link between robust statistics and deep learning through f-GAN (Figure 3). Huber’s contamination model. On the other hand, for covariance matrix estimation, a two-hidden-layer structure seems to be necessary. An

To this question, we have already known that

What choices of

That is, Huber’s contamination model? How to characterize the class of network structures that lead to rate-optimal robust procedures under Huber’s contamination model? how to specify an appropriate neural network architecture that

It turns out the structure of

Indeed shows a roughly linear dependence on

To this question, we have already known that

What is the smallest mini-batch size that guarantees robustness?

..."
Remarkably, it is robust because that is, Huber’s is leads to rate-optimal robust procedures under Huber’s contamination model. On the other hand, for covariance matrices, results also include the behavior of error against dimension. When

Each of the above points will lead to a nontrivial research problem in computational robust statistics. Moreover, we also know that Kullback-Leibler learning (MLE) is not robust, because That is, Huber’s

As we have just mentioned, even an approximate algorithm for optimizing Tukey’s depth can be derived from basic f-divergence inequalities. This is confirmed in Figure 18.

The relation \( f \) can be derived from basic f-divergence inequalities \[ f(x) = \frac{1}{\lambda} \log(1 + \lambda x) \] for \( \lambda > 0 \). This is confirmed in Figure 18.

However, this is not against

As we have just mentioned, even an approximate algorithm for optimizing Tukey’s depth can be derived from basic f-divergence inequalities \[ f(x) = \frac{1}{\lambda} \log(1 + \lambda x) \] for \( \lambda > 0 \). This is confirmed in Figure 18.

What choices of \( f \)-GAN do not lose under projective angle to study the robustness of various f-GAN procedures. Our research question is: Moreover, we also know that Kullback-Leibler learning (MLE) is not robust, because that is, Huber’s

\[ \hat{\theta} = \arg\min_{\eta \in \mathbb{R}^p} \max_{T \in \mathcal{T}} \left[ \frac{1}{n} \sum_{i=1}^n \log T(X_i) + E_{\eta} \log(1 - T(X)) \right] + \log 4 \]

\[ X_1, \ldots, X_n \sim (1 - \epsilon)N(\theta, I_p) + \epsilon N(\tilde{\theta}, I_p) \]

\[ \hat{\theta} \approx (1 - \epsilon)\theta + \epsilon \tilde{\theta} \]

\[ \hat{\theta} \approx \theta \]

\[ \hat{\theta} \approx \theta \]
JS-GAN

A classifier with hidden layers leads to robustness. Why?
A classifier with hidden layers leads to robustness. Why?

\[ JS_g(P, Q) = \max_{w \in \mathbb{R}^d} \left[ P \log \frac{1}{1 + e^{-w^T g(X)}} + Q \log \frac{1}{1 + e^{w^T g(X)}} \right] + \log 4. \]
A classifier with hidden layers leads to robustness. Why?

\[ JS_g(\mathbb{P}, \mathbb{Q}) = \max_{w \in \mathbb{R}^d} \left[ \mathbb{P} \log \frac{1}{1 + e^{-w^T g(X)}} + \mathbb{Q} \log \frac{1}{1 + e^{w^T g(X)}} \right] + \log 4. \]

**Proposition.**

\[ JS_g(\mathbb{P}, \mathbb{Q}) = 0 \iff \mathbb{P} g(X) = \mathbb{Q} g(X) \]
JS-GAN

\[ \widehat{\theta} = \arg\min_{\eta \in \mathbb{R}^p} \max_{T \in \mathcal{T}} \left[ \frac{1}{n} \sum_{i=1}^{n} \log T(X_i) + E_{\eta} \log(1 - T(X)) \right] + \log 4 \]

**Theorem [GLYZ18].** For a neural network class \( \mathcal{T} \) with at least one hidden layer and appropriate regularization, we have

\[
\| \widehat{\theta} - \theta \|^2 \lesssim \begin{cases} 
\frac{p}{n} + \epsilon^2 & \text{(indicator/sigmoid/ramp)} \\
\frac{p \log p}{n} + \epsilon^2 & \text{(ReLU}s+sigmoid features}
\end{cases}
\]

with high probability uniformly over \( \theta \in \mathbb{R}^p, Q \).
JS-GAN: Adaptation to Unknown Covariance

unknown covariance?

\[ X_1, \ldots, X_n \sim (1 - \epsilon)N(\theta, \Sigma) + \epsilon Q \]
JS-GAN: Adaptation to Unknown Covariance

unknown covariance?

\[ X_1, \ldots, X_n \sim (1 - \epsilon)N(\theta, \Sigma) + \epsilon Q \]

\[
(\hat{\theta}, \hat{\Sigma}) = \arg\min_{\theta, \Sigma} \max_{T \in \mathcal{T}} \left[ \frac{1}{n} \sum_{i=1}^{n} \log T(X_i) + \mathbb{E}_{X \sim N(\eta, \Gamma)} \log(1 - T(X)) \right]
\]
JS-GAN: Adaptation to Unknown Covariance

unknown covariance?

\[ X_1, \ldots, X_n \sim (1 - \epsilon)N(\theta, \Sigma) + \epsilon Q \]

\[
(\hat{\theta}, \hat{\Sigma}) = \arg\min_{\eta, \Gamma} \max_{T \in \mathcal{T}} \left[ \frac{1}{n} \sum_{i=1}^{n} \log T(X_i) + \mathbb{E}_{X \sim N(\eta, \Gamma)} \log(1 - T(X)) \right]
\]

no need to change the discriminator class
Generalization

Strong Contamination model:

\[ X_1, \ldots, X_n \overset{iid}{\sim} P \text{ for some } P \text{ satisfying } TV(P, E(\theta, \Sigma, H)) \leq \epsilon \]

\[
(\hat{\theta}, \hat{\Sigma}, \hat{H}) = \arg\min_{\eta \in \mathbb{R}^p, \Gamma \in \mathcal{E}_p(M), H \in \mathcal{H}(M')} \max_{T \in \mathcal{T}} \left[ \frac{1}{n} \sum_{i=1}^{n} S(T(X_i), 1) + \mathbb{E}_{X \sim E(\eta, \Gamma, G)} S(T(X), 0) \right]
\]

A scoring rule \( S \) is regular if both \( S(\cdot, 0) \) and \( S(\cdot, 1) \) are real-valued, except possibly that \( S(0, 1) = -\infty \) or \( S(1, 0) = -\infty \). The celebrated Savage representation [50] asserts that a regular scoring rule \( S \) is proper if and only if there is a convex function \( G(\cdot) \), such that

\[
\begin{aligned}
S(t, 1) &= G(t) + (1 - t)G'(t), \\
S(t, 0) &= G(t) - tG'(t).
\end{aligned}
\] (10)

Here, \( G'(t) \) is a subgradient of \( G \) at the point \( t \). Moreover, the statement also holds for strictly proper scoring rules when convex is replaced by strictly convex.
Consistency

**Theorem [GYZ19].** For a neural network class $\mathcal{T}$ with at least one hidden layer and appropriate regularization, we have

\[
\|\hat{\theta} - \theta\|^2 \leq C \left( \frac{p}{n} \vee \epsilon^2 \right), \\
\|\hat{\Sigma} - \Sigma\|_{\text{op}}^2 \leq C \left( \frac{p}{n} \vee \epsilon^2 \right),
\]
Example 1: Log Score and JS-GAN

1. **Log Score.** The log score is perhaps the most commonly used rule because of its various intriguing properties [31]. The scoring rule with \( S(t, 1) = \log t \) and \( S(t, 0) = \log(1 - t) \) is regular and strictly proper. Its Savage representation is given by the convex function \( G(t) = t \log t + (1 - t) \log(1 - t) \), which is interpreted as the negative Shannon entropy of Bernoulli\( (t) \). The corresponding divergence function \( D_T(P, Q) \), according to Proposition 3.1, is a variational lower bound of the Jensen-Shannon divergence

\[
\text{JS}(P, Q) = \frac{1}{2} \int \log \left( \frac{dP}{dP + dQ} \right) dP + \frac{1}{2} \int \log \left( \frac{dQ}{dP + dQ} \right) dQ + \log 2.
\]

Its sample version (13) is the original GAN proposed by [25] that is widely used in learning distributions of images.
Example 2: Zero-One Score and TV-GAN

2. Zero-One Score. The zero-one score $S(t, 1) = 2\mathbb{I}\{t \geq 1/2\}$ and $S(t, 0) = 2\mathbb{I}\{t < 1/2\}$ is also known as the misclassification loss. This is a regular proper scoring rule but not strictly proper. The induced divergence function $D_T(P, Q)$ is a variational lower bound of the total variation distance

$$TV(P, Q) = P\left(\frac{dP}{dQ} \geq 1\right) - Q\left(\frac{dP}{dQ} \geq 1\right) = \frac{1}{2} \int |dP - dQ|.$$ 

The sample version (13) is recognized as the TV-GAN that is extensively studied by [21] in the context of robust estimation.
Example 3: Quadratic Score and LS-GAN

3. Quadratic Score. Also known as the Brier score [6], the definition is given by $S(t, 1) = -(1 - t)^2$ and $S(t, 0) = -t^2$. The corresponding convex function in the Savage representation is given by $G(t) = -t(1 - t)$. By Proposition 2.1, the divergence function (3) induced by this regular strictly proper scoring rule is a variational lower bound of the following divergence function,

$$
\Delta(P, Q) = \frac{1}{8} \int \frac{(dP - dQ)^2}{dP + dQ},
$$

known as the triangular discrimination. The sample version (5) belongs to the family of least-squares GANs proposed by [39].
4. **Boosting Score.** The boosting score was introduced by [7] with \( S(t, 1) = -\left( \frac{1-t}{t} \right)^{1/2} \) and \( S(t, 0) = -\left( \frac{t}{1-t} \right)^{1/2} \) and has an connection to the AdaBoost algorithm. The corresponding convex function in the Savage representation is given by \( G(t) = -2\sqrt{t(1-t)} \). The induced divergence function \( D_T(P, Q) \) is thus a variational lower bound of the squared Hellinger distance

\[
H^2(P, Q) = \frac{1}{2} \int \left( \sqrt{dP} - \sqrt{dQ} \right)^2.
\]
Example 5: Beta Score and new GANs

5. Beta Score. A general Beta family of proper scoring rules was introduced by [7] with $S(t, 1) = -\int_t^1 c^{\alpha-1}(1 - c)^\beta dc$ and $S(t, 0) = -\int_0^t c^\alpha(1 - c)^{\beta-1} dc$ for any $\alpha, \beta > -1$. The log score, the quadratic score and the boosting score are special cases of the Beta score with $\alpha = \beta = 0$, $\alpha = \beta = 1$, $\alpha = \beta = -1/2$. The zero-one score is a limiting case of the Beta score by letting $\alpha = \beta \to \infty$. Moreover, it also leads to asymmetric scoring rules with $\alpha \neq \beta$. 
Robust Learning of Gaussian Distributions

<table>
<thead>
<tr>
<th>$Q$</th>
<th>$n$</th>
<th>$p$</th>
<th>$\epsilon$</th>
<th>TV-GAN</th>
<th>JS-GAN</th>
<th>Dimension Halving</th>
<th>Iterative Filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N(0.5 \cdot 1_p, I_p)$</td>
<td>50,000</td>
<td>100</td>
<td>.2</td>
<td><strong>0.0953 (0.0064)</strong></td>
<td>0.1144 (0.0154)</td>
<td>0.3247 (0.0058)</td>
<td>0.1472 (0.0071)</td>
</tr>
<tr>
<td>$N(0.5 \cdot 1_p, I_p)$</td>
<td>5,000</td>
<td>100</td>
<td>.2</td>
<td><strong>0.1941 (0.0173)</strong></td>
<td>0.2182 (0.0527)</td>
<td>0.3568 (0.0197)</td>
<td>0.2285 (0.0103)</td>
</tr>
<tr>
<td>$N(0.5 \cdot 1_p, I_p)$</td>
<td>50,000</td>
<td>200</td>
<td>.2</td>
<td><strong>0.1108 (0.0093)</strong></td>
<td>0.1573 (0.0815)</td>
<td>0.3251 (0.0078)</td>
<td>0.1525 (0.0045)</td>
</tr>
<tr>
<td>$N(0.5 \cdot 1_p, I_p)$</td>
<td>50,000</td>
<td>100</td>
<td>.05</td>
<td>0.0913 (0.0527)</td>
<td>0.1390 (0.0050)</td>
<td>0.0814 (0.0056)</td>
<td><strong>0.0530 (0.0052)</strong></td>
</tr>
<tr>
<td>$N(5 \cdot 1_p, I_p)$</td>
<td>50,000</td>
<td>100</td>
<td>.2</td>
<td>2.7721 (0.1285)</td>
<td><strong>0.0534 (0.0041)</strong></td>
<td>0.3229 (0.0087)</td>
<td>0.1471 (0.0059)</td>
</tr>
<tr>
<td>$N(0.5 \cdot 1_p, \Sigma)$</td>
<td>50,000</td>
<td>100</td>
<td>.2</td>
<td>0.1189 (0.0195)</td>
<td><strong>0.1148 (0.0234)</strong></td>
<td>0.3241 (0.0088)</td>
<td>0.1426 (0.0113)</td>
</tr>
<tr>
<td>Cauchy$(0.5 \cdot 1_p)$</td>
<td>50,000</td>
<td>100</td>
<td>.2</td>
<td>0.0738 (0.0053)</td>
<td><strong>0.0525 (0.0029)</strong></td>
<td>0.1045 (0.0071)</td>
<td>0.0633 (0.0042)</td>
</tr>
</tbody>
</table>

Table: Comparison of various robust mean estimation methods. The smallest error of each case is highlighted in bold.

- **Dimension Halving:** [Lai et al.’16]

- **Iterative Filtering:** [Diakonikolas et al.’17]
Robust Learning of Cauchy Distributions

Table 4: Comparison of various methods of robust location estimation under Cauchy distributions. Samples are drawn from \((1 - \epsilon)\text{Cauchy}(0, I_p) + \epsilon Q\) with \(\epsilon = 0.2, p = 50\) and various choices of \(Q\). Sample size: 50,000. Discriminator net structure: 50-50-25-1. Generator \(g_\omega(\xi)\) structure: 48-48-32-24-12-1 with absolute value activation function in the output layer.

<table>
<thead>
<tr>
<th>Contamination (Q)</th>
<th>JS-GAN ((G_1))</th>
<th>JS-GAN ((G_2))</th>
<th>Dimension Halving</th>
<th>Iterative Filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cauchy((1.5 \times 1_p, I_p))</td>
<td><strong>0.0664 (0.0065)</strong></td>
<td>0.0743 (0.0103)</td>
<td>0.3529 (0.0543)</td>
<td>0.1244 (0.0114)</td>
</tr>
<tr>
<td>Cauchy((5.0 \times 1_p, I_p))</td>
<td><strong>0.0480 (0.0058)</strong></td>
<td>0.0540 (0.0064)</td>
<td>0.4855 (0.0616)</td>
<td>0.1687 (0.0310)</td>
</tr>
<tr>
<td>Cauchy((1.5 \times 1_p, 5 \times I_p))</td>
<td>0.0754 (0.0135)</td>
<td><strong>0.0742 (0.0111)</strong></td>
<td>0.3726 (0.0530)</td>
<td>0.1220 (0.0112)</td>
</tr>
<tr>
<td>Normal((1.5 \times 1_p, 5 \times I_p))</td>
<td><strong>0.0702 (0.0064)</strong></td>
<td>0.0713 (0.0088)</td>
<td>0.3915 (0.0232)</td>
<td>0.1048 (0.0288)</td>
</tr>
</tbody>
</table>

- **Dimension Halving**: [Lai et al.’16]

- **Iterative Filtering**: [Diakonikolas et al.’17]
Discriminator identifies outliers

\[
(1 - \epsilon)N(0_p, I_p) + \epsilon Q
\]

\[
N(5 * 1_p, I_p)
\]

- Discriminator helps identify outliers or contaminated samples
- Generator fits uncontaminated portion of true samples
Application: Price of 50 stocks from 2007/01 to 2018/12
Corps are selected by ranking in market capitalization
Log-return. $y[i] = \log(\text{price}_{i+1}/\text{price}_{i})$
Fit data by Elliptica-GAN.
Apply SVD on scatter.
Dimension reduction on $\mathbb{R}^2$.
Outlier $x$ and $o$ are selected from Discriminator value distribution.
Discriminator value distribution from (Elliptical) Generator and real samples. Outliers are chosen from samples larger/ lower than a chosen percentile of Generator distribution.
Loading of PCA.
First two direction are dominated by few corps —> not robust
Loading of Elliptical Scatter:
Comparing with PCA, it’s more robust in the sense that it does not totally dominate by Financial company (JPM, GS)
Reference


Thank You