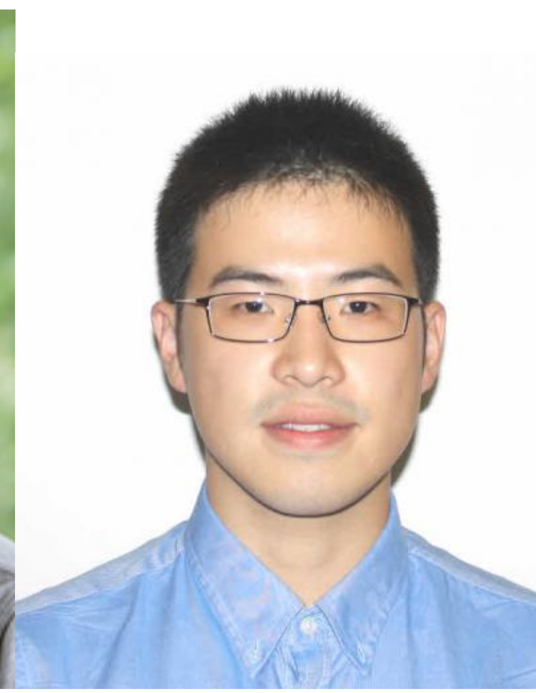
Robust Statistics and Generative Adversarial Networks

Yuan YAO
HKUST

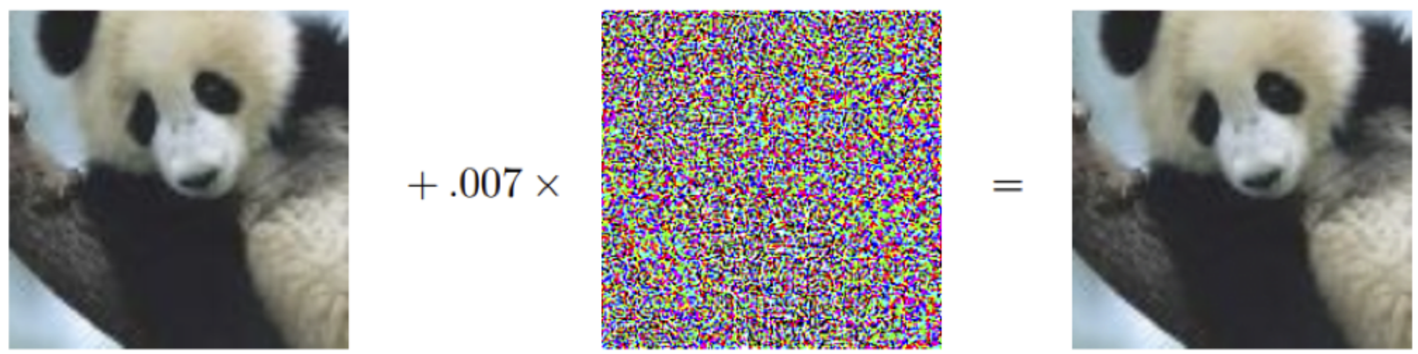
Chao Gao (Chicago)



Jiyu Liu (Yale)

Weizhi Zhu (HKUST)

Deep Learning is Notoriously Not Robust!

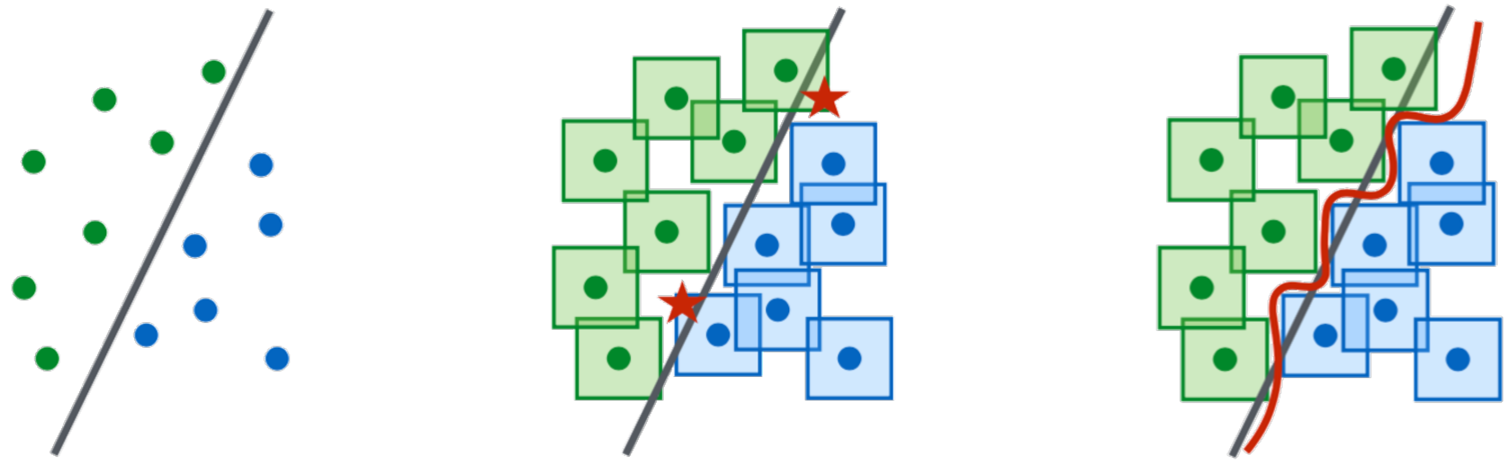


$\boldsymbol{x}$	$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$	$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"panda"	"nematode"	"gibbon"
57.7% confidence	8.2% confidence	99.3 % confidence

[Goodfellow et al., 2014]

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

Robust Optimization



- Traditional training:

$$\min_{\theta} J_n(\theta, \mathbf{z} = (x_i, y_i)_{i=1}^n)$$

  - e.g. square or cross-entropy loss as negative log-likelihood of logit models

- Robust optimization (Madry et al. ICLR'2018):

$$\min_{\theta} \max_{\|\epsilon_i\| \leq \delta} J_n(\theta, \mathbf{z} = (x_i + \epsilon_i, y_i)_{i=1}^n)$$

  - robust to any distributions, yet computationally hard

Distributionally Robust Optimization (DRO)

- Distributional Robust Optimization:

$$\min_{\theta} \max_{\epsilon} \mathbb{E}_{\mathbf{z} \sim P_\epsilon \in \mathcal{D}}[J_n(\theta, \mathbf{z})]$$

- $\mathcal{D}$ 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

Wasserstein DRO and Sqrt-Lasso

Theorem (B., Kang, Murthy (2016)) Suppose that

$$c\left((x,y),(x',y')\right) = \begin{cases} \|x-x'\|_q^2 & \text{if} \quad y=y' \\ \infty & \text{if} \quad y \neq y' \end{cases}.$$

Then, if $1/p + 1/q = 1$

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

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

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

Then for

$$P_n' = \frac{1}{n}\sum_i \delta_{x_i'}$$

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 δ and well-separated x's. Sqrt-Lasso

$$\min_\beta \left\{ E_{P_n}^{1/2}\left[\left(Y - \beta^T X\right)^2\right] + \sqrt{\delta}\|\beta\|_1 \right\}^2$$

$$= \min_\beta \max_{P: D_c(P, P_n) \leq \delta} E_P\left(\left(Y - \beta^T X\right)^2\right)$$

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?

$$\mathcal{D} = \{P_\epsilon : TV(P_\epsilon, \textsf{uniform distribution}) \leq \epsilon\}?$$

Huber's Model

$$X_1, ..., X_n \sim (1 - \epsilon)P_\theta + \epsilon Q$$

[Huber 1964]

Huber's Model

$$X_1, ..., X_n \sim (1-\epsilon)P_\theta + \epsilon Q$$

parameter of interest

[Huber 1964]

Huber's Model

$$X_1, ..., X_n \sim (1 - \epsilon)P_\theta + \epsilon Q$$

contamination proportion

parameter of interest

[Huber 1964]

Huber's Model

$$X_1, ..., X_n \sim (1-\epsilon)P_\theta + \epsilon Q$$

contamination proportion

arbitrary contamination

parameter of interest

[Huber 1964]

An Example

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

An Example

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

how to estimate ?

Robust Maxmum-Likelihood Does not work!

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

Robust Maxmum-Likelihood Does not work!

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

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

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

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**

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

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

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

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

Multivariate Location Depth

$$\left\{\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\{u^T X_i > u^T \eta\} \wedge \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\{u^T X_i \leq u^T \eta\}\right\}$$

Multivariate Location Depth

$$\min_{\|u\|=1} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i > u^T \eta\} \wedge \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{u^T X_i \leq u^T \eta\} \right\}$$

Multivariate Location Depth

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

Multivariate Location Depth

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

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

[Tukey, 1975]

Regression Depth

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

Regression Depth

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

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

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

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

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

Regression Depth

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

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

projection $$u^TXy|X \sim N(u^TXX^T\beta, \sigma^2 u^TXX^Tu)$$

$$\hat{\beta} = \underset{\eta\in\mathbb{R}^p}{\operatorname{argmax}} \min_{u\in\mathbb{R}^p} \left\{ \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{u^TX_i(y_i - X_i^T\eta) > 0\} \wedge \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{u^TX_i(y_i - X_i^T\eta) \leq 0\} \right\}$$

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 $u^TXy|X \sim N(u^TXX^T\beta, \sigma^2 u^TXX^Tu)$

$$\hat{\beta} = \underset{\eta\in\mathbb{R}^p}{\operatorname{argmax}} \min_{u\in\mathbb{R}^p} \left\{ \frac{1}{n}\sum_{i=1}^n \mathbb{I}\{u^TX_i(y_i - X_i^T\eta) > 0\} \wedge \frac{1}{n}\sum_{i=1}^n \mathbb{I}\{u^TX_i(y_i - X_i^T\eta) \leq 0\} \right\}$$

[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:

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

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:

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

empirical version:

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

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:

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

empirical version:

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

[Mizera, 2002]

Multi-task Regression Depth

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

Multi-task Regression Depth

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

$p = 1, X = 1 \in \mathbb{R},$

$$\mathcal{D}_{\mathcal{U}}(b, \mathbb{P}) = \inf_{u \in \mathcal{U}} \mathbb{P}\left\{u^T (Y - b) \geq 0\right\}$$

Multi-task Regression Depth

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

$p = 1, X = 1 \in \mathbb{R},$

$$\mathcal{D}_{\mathcal{U}}(b,\mathbb{P}) = \inf_{u\in\mathcal{U}} \mathbb{P}\left\{u^T(Y - b) \geq 0\right\}$$

$m = 1,$

$$\mathcal{D}_{\mathcal{U}}(\beta,\mathbb{P}) = \inf_{U\in\mathcal{U}} \mathbb{P}\left\{u^T X(y - \beta^T X) \geq 0\right\}$$

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 $1 - 2\delta$.

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

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

$$(X_1, Y_1), ..., (X_n, Y_n) \sim (1 - \epsilon)P_B + \epsilon Q$$

Multi-task Regression Depth

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

$$(X_1, Y_1), ..., (X_n, Y_n) \sim (1-\epsilon) P_B + \epsilon Q$$

Theorem [G17]. For some $C > 0$,

$$\mathsf{Tr}((\widehat{B} - B)^T \Sigma (\widehat{B} - B)) \leq C\sigma^2 \left(\frac{pm}{n} \vee \epsilon^2 \right),$$

$$\|\widehat{B} - B\|_{\mathrm{F}}^2 \leq C \frac{\sigma^2}{\kappa^2} \left(\frac{pm}{n} \vee \epsilon^2 \right),$$

with high probability uniformly over B, Q.

Covariance Matrix

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

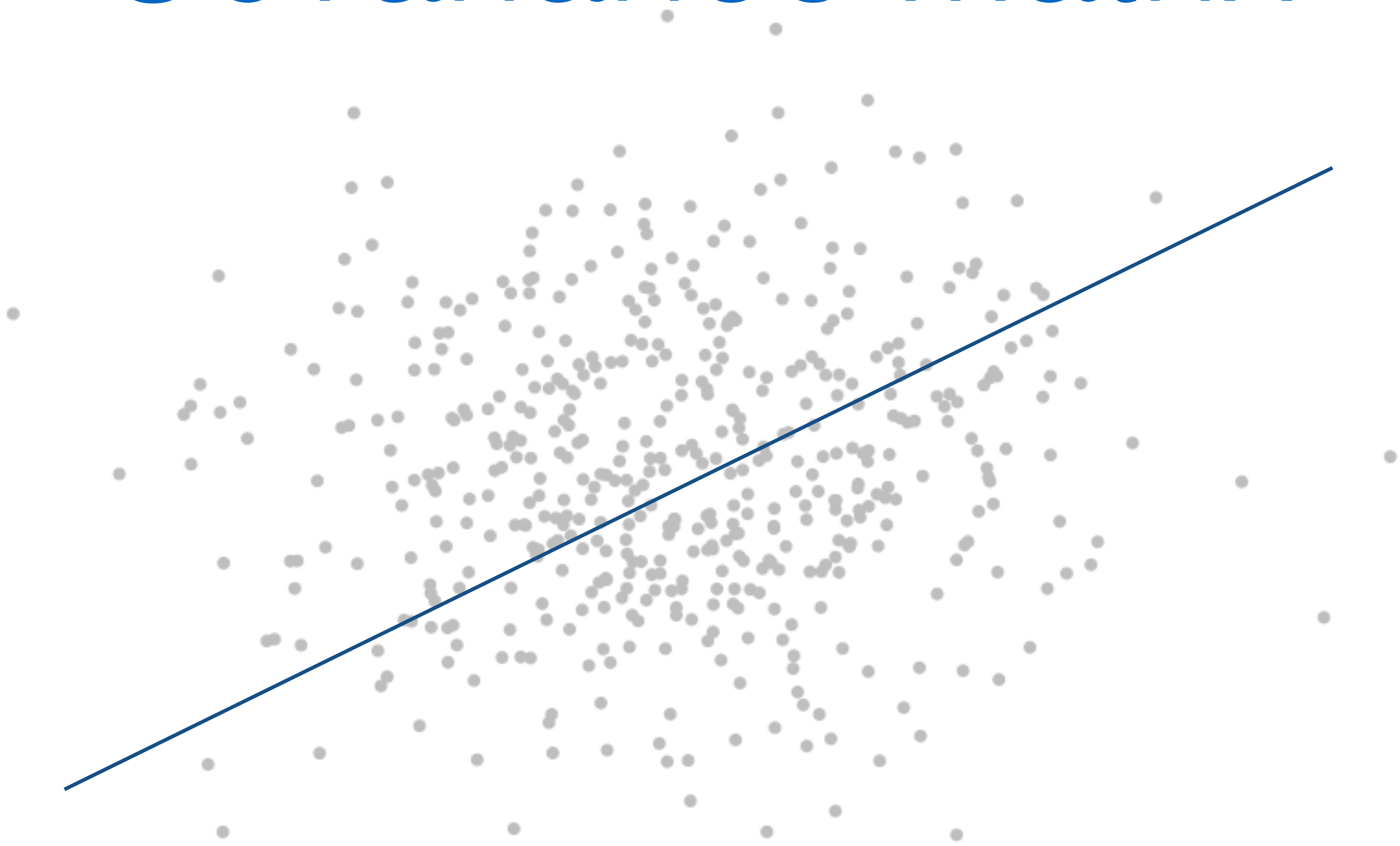
Covariance Matrix

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

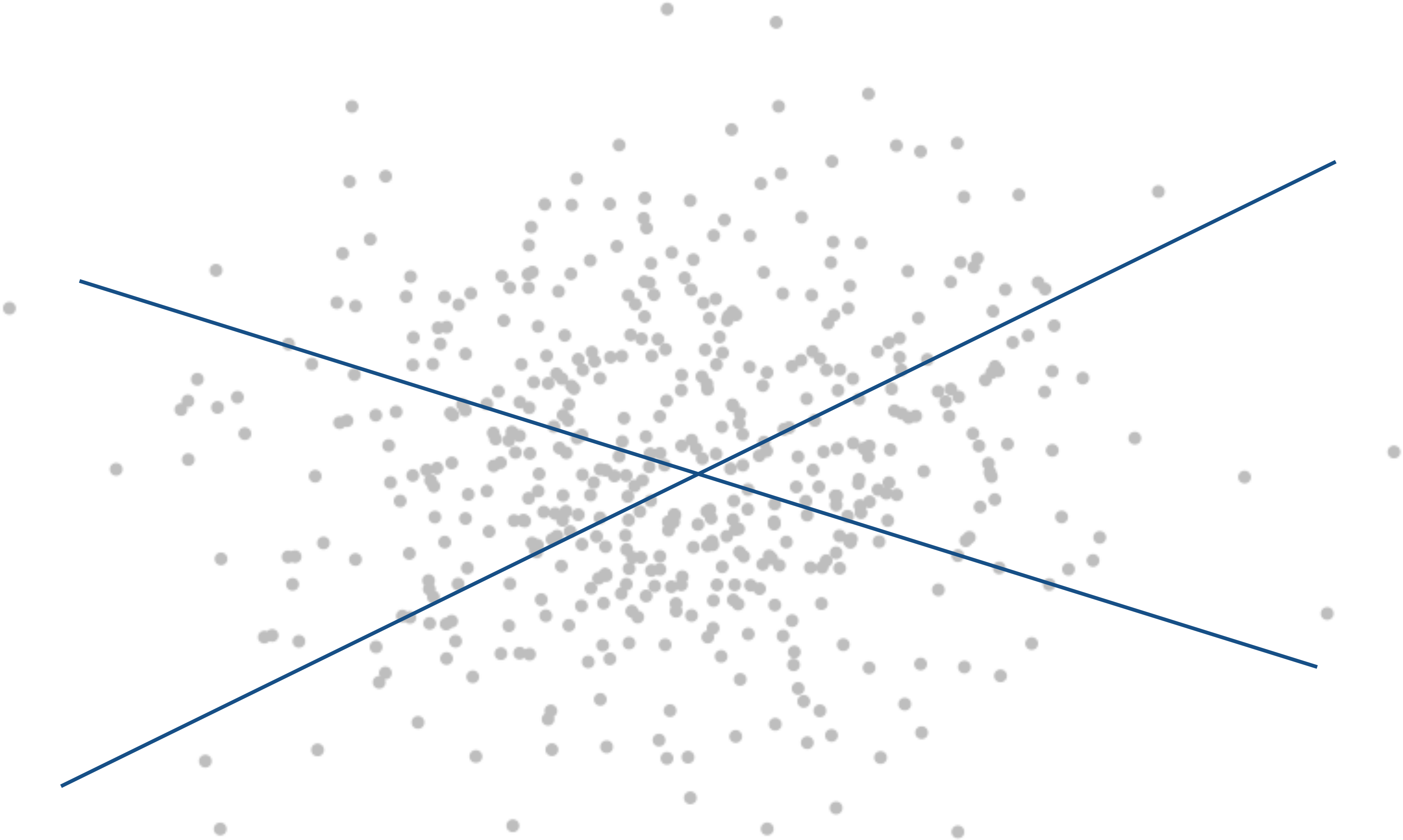
how to estimate ?

Covariance Matrix

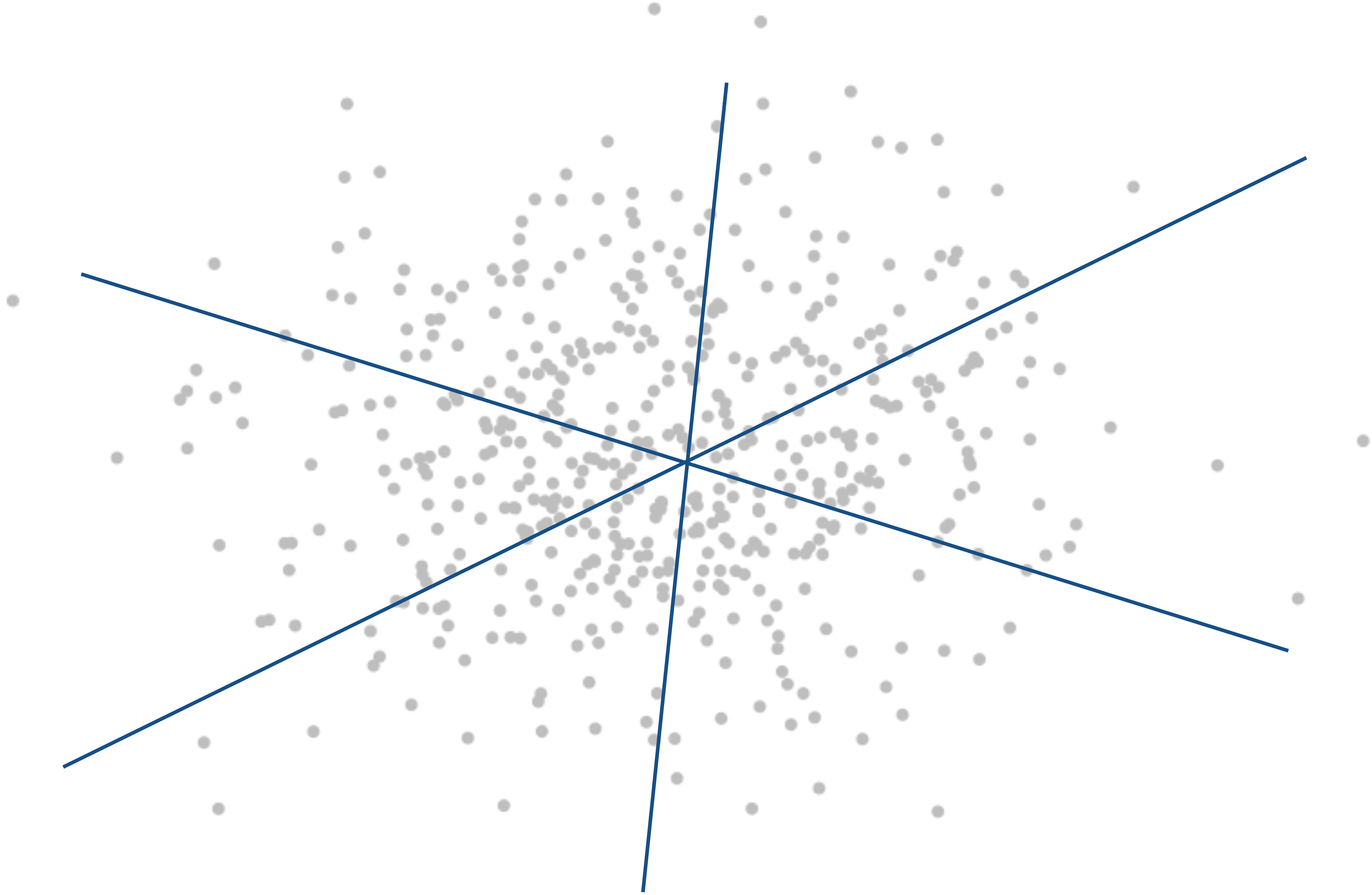
Covariance Matrix



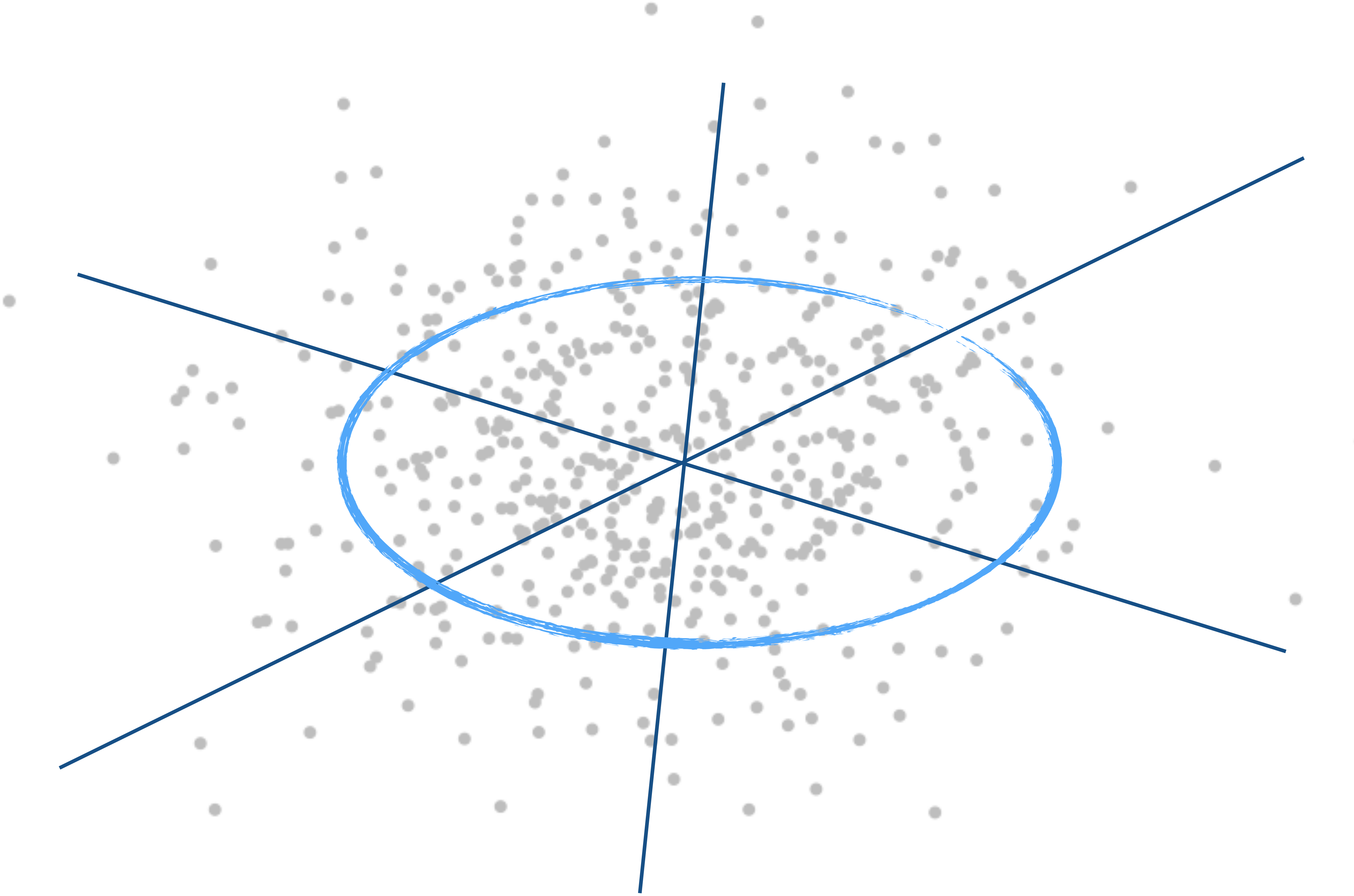
Covariance Matrix



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

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

$$\hat{\Gamma} = \arg\max_{\Gamma \succeq 0} \mathcal{D}(\Gamma, \{X_i\}_{i=1}^n) \qquad \hat{\Sigma} = \hat{\Gamma}/\beta$$

Covariance Matrix

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

$$\hat{\Gamma} = \arg\max_{\Gamma \succeq 0} \mathcal{D}(\Gamma, \{X_i\}_{i=1}^n) \qquad \hat{\Sigma} = \hat{\Gamma}/\beta$$

Theorem [CGR15]. For some $C > 0$,

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

with high probability uniformly over Σ, Q .

Summary

mean	$\|\cdot\|^2$	$\frac{p}{n} \vee \epsilon^2$
reduced rank regression	$\|\cdot\|_{\mathrm{F}}^2$	$\frac{\sigma^2}{\kappa^2}\frac{r(p+m)}{n} \vee \frac{\sigma^2}{\kappa^2}\epsilon^2$
Gaussian graphical model	$\|\cdot\|_{\ell_1}^2$	$\frac{s^2\log(ep/s)}{n} \vee s\epsilon^2$
covariance matrix	$\|\cdot\|_{\mathrm{op}}^2$	$\frac{p}{n} \vee \epsilon^2$
sparse PCA	$\|\cdot\|_{\mathrm{F}}^2$	$\frac{s\log(ep/s)}{n\lambda^2} \vee \frac{\epsilon^2}{\lambda^2}$

Summary

mean	$\|\cdot\|^2$	$\frac{p}{n} \vee \epsilon^2$
reduced rank regression	$\|\cdot\|_{\mathrm{F}}^2$	$\frac{\sigma^2}{\kappa^2}\frac{r(p+m)}{n} \vee \frac{\sigma^2}{\kappa^2}\epsilon^2$
Gaussian graphical model	$\|\cdot\|_{\ell_1}^2$	$\frac{s^2\log(ep/s)}{n} \vee s\epsilon^2$
covariance matrix	$\|\cdot\|_{\mathrm{op}}^2$	$\frac{p}{n} \vee \epsilon^2$
sparse PCA	$\|\cdot\|_{\mathrm{F}}^2$	$\frac{s\log(ep/s)}{n\lambda^2} \vee \frac{\epsilon^2}{\lambda^2}$

Computation

Computational Challenges

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

Computational Challenges

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

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

Computational Challenges

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

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 ϵ

Advantages of Tukey Median

Advantages of Tukey Median

- **A well-defined objective function**

Advantages of Tukey Median

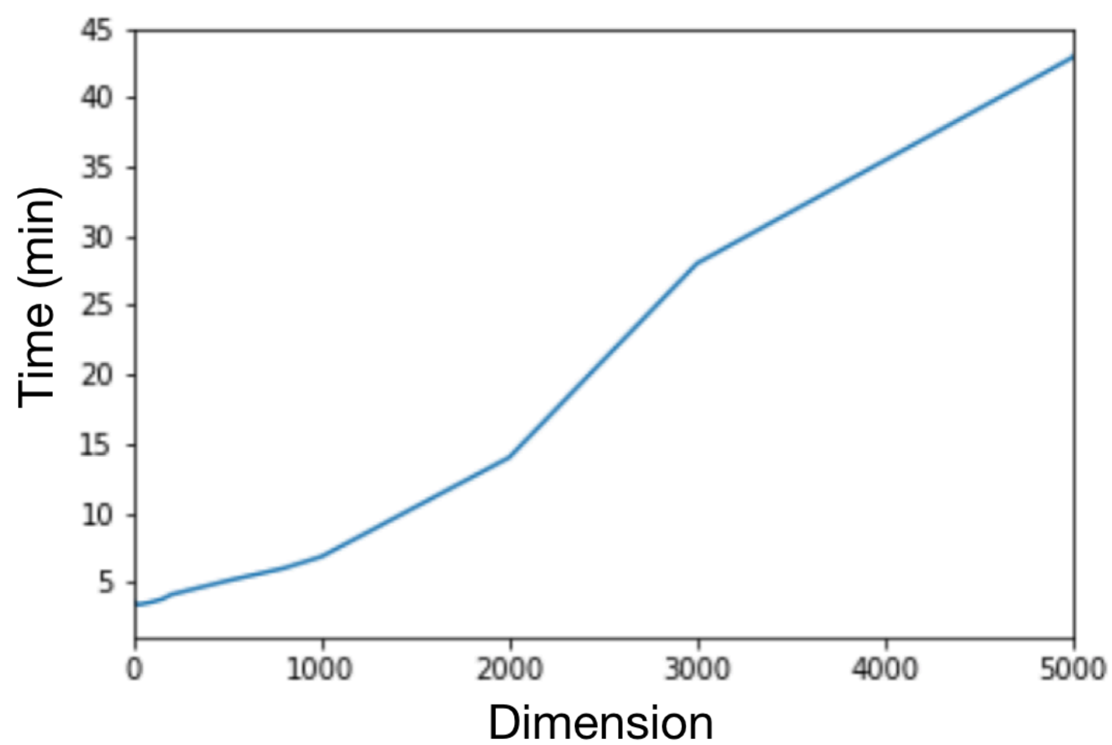
- **A well-defined objective function**
- **Adaptive to ϵ and Σ**

Advantages of Tukey Median

- **A well-defined objective function**
- **Adaptive to ϵ and Σ**
- **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.

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

- *Dimension Halving*: [Lai et al.'16] `https://github.com/kal2000/AgnosticMeanAndCovarianceCode`.
- *Iterative Filtering*: [Diakonikolas et al.'17] `https://github.com/hoonose/robust-filter`.

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

$$D_f(P\|Q) = \int f\left(\frac{p}{q}\right) dQ. \tag{8}$$

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

$$D_f(P\|Q) \geq \sup_{T\in\mathcal{T}} \left[\mathbb{E}_P T(X) - \mathbb{E}_Q f^*(T(X))\right]. \tag{9}$$

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)

$$\widehat{P} = \arg\min_{Q\in\mathcal{Q}} \sup_{T\in\mathcal{T}} \left[\frac{1}{n}\sum_{i=1}^{n} T(X_i) - \mathbb{E}_Q f^*(T(X))\right]. \tag{10}$$

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

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

Consider the special case

$$\mathcal{T} = \left\{ f'\left(\frac{\widetilde{q}}{q}\right) : \widetilde{q} \in \widetilde{\mathcal{Q}} \right\}. \tag{11}$$

which is tight if $P \in \widetilde{\mathcal{Q}}$. The sample version leads to the following f-learning

$$\widehat{P} = \arg\min_{Q \in \mathcal{Q}} \sup_{\widetilde{Q} \in \widetilde{\mathcal{Q}}} \left[\frac{1}{n} \sum_{i=1}^{n} f'\left(\frac{\widetilde{q}(X_i)}{q(X_i)} \right) - \mathbb{E}_Q f^*\left(f'\left(\frac{\widetilde{q}(X)}{q(X)} \right) \right) \right]. \tag{12}$$

- If $f(x) = x \log x$, $\mathcal{Q} = \widetilde{\mathcal{Q}}$, (12) $\Rightarrow$ Maximum Likelihood Estimate
- If $f(x) = (x-1)+$, then $D_f(P\|Q) = \frac{1}{2}\int |p-q|$ is the TV-distance, $f^*(t) = t\mathbb{I}\{0 \le t \le 1\}$, f-GAN $\Rightarrow$ TV-GAN
  - $\mathcal{Q} = \{N(\eta, I_p) : \eta \in \mathbb{R}^p\}$ and $\widetilde{\mathcal{Q}} = \{N(\widetilde{\eta}, I_p) : \|\widetilde{\eta} - \eta\| \le r\}$, (12) $\overset{r \to 0}{\Rightarrow}$ Tukey's Median

f-Learning

f-Learning

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

f-Learning

f-divergence

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

$$f(u) = \sup_t (tu - f^*(t))$$

f-Learning

f-divergence

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

variational representation

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

f-Learning

f-divergence

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

variational representation

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

optimal T

$$T(x) = f'\left(\frac{p(x)}{q(x)}\right)$$

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

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

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

f-Learning

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

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

f-Learning

f-GAN

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

f-Learning

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

f-Learning

f-GAN

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

f-Learning

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

[Nowozin, Cseke, Tomioka]

f-Learning

f-Learning

Jensen-Shannon	$f(x) = x \log x - (x+1)\log(x+1)$	**GAN**

[Goodfellow et al.

f-Learning

Jensen-Shannon	$f(x) = x \log x - (x+1)\log(x+1)$	**GAN**
Kullback-Leibler	$f(x) = x \log x$	**MLE**

[Goodfellow et al.

f-Learning

Jensen-Shannon	$f(x) = x \log x - (x+1)\log(x+1)$	**GAN**
Kullback-Leibler	$f(x) = x \log x$	**MLE**
Hellinger Squared	$f(x) = 2 - 2\sqrt{x}$	**rho**

[Goodfellow et al., Baraud and Birge]

f-Learning

Jensen-Shannon	$f(x) = x \log x - (x+1)\log(x+1)$	**GAN**
Kullback-Leibler	$f(x) = x \log x$	**MLE**
Hellinger Squared	$f(x) = 2 - 2\sqrt{x}$	**rho**
Total Variation	$f(x) = (x-1)_+$	**depth**

[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

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

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

TV-Learning

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

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

$r \to 0$

TV-Learning

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

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

$r\to 0$

Tukey depth $\displaystyle\max_{\theta\in\mathbb{R}}\min_{||u||=1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\left\{u^TX_i\geq u^T\theta\right\}$

TV-Learning

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

TV-Learning

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

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

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

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

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

$r\to 0$

(related to) matrix depth

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

robust statistics community

deep learning community

robust statistics community

f-Learning

f-GAN

deep learning community

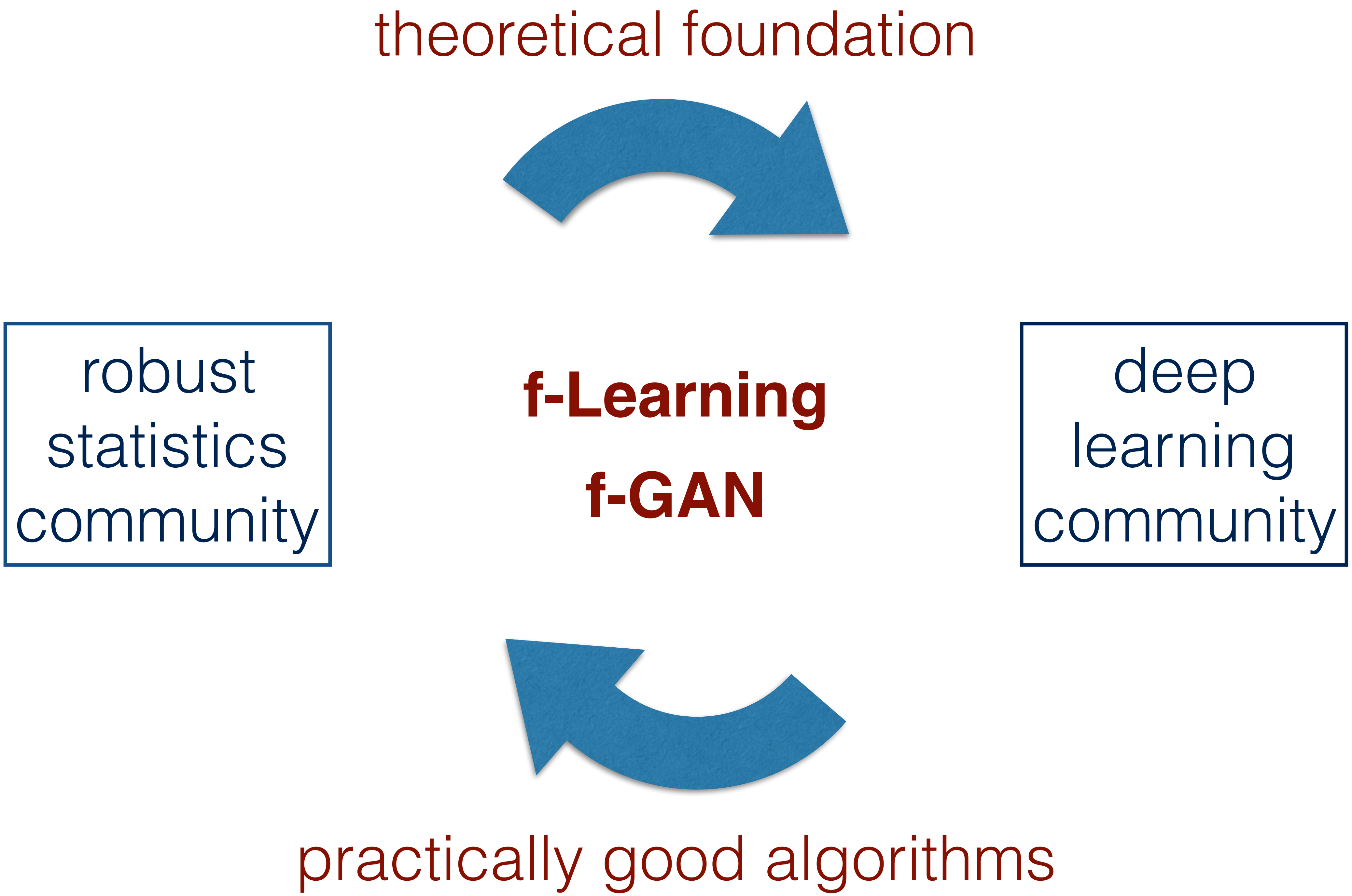
robust statistics community

f-Learning

f-GAN

deep learning community

practically good algorithms


theoretical foundation
robust statistics community
f-Learning
f-GAN
deep learning community
practically good algorithms

TV-GAN

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

TV-GAN

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

$$N(\eta, I_p)$$

TV-GAN

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

$N(\eta, I_p)$

logistic regression classifier

TV-GAN

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

$N(\eta, I_p)$

logistic regression classifier

Theorem [GLYZ18]. For some $C > 0$,

$$\|\widehat{\theta} - \theta\|^2 \leq C \left(\frac{p}{n} \vee \epsilon^2 \right)$$

with high probability uniformly over $\theta \in \mathbb{R}^p, Q$.

TV-GAN

rugged landscape!

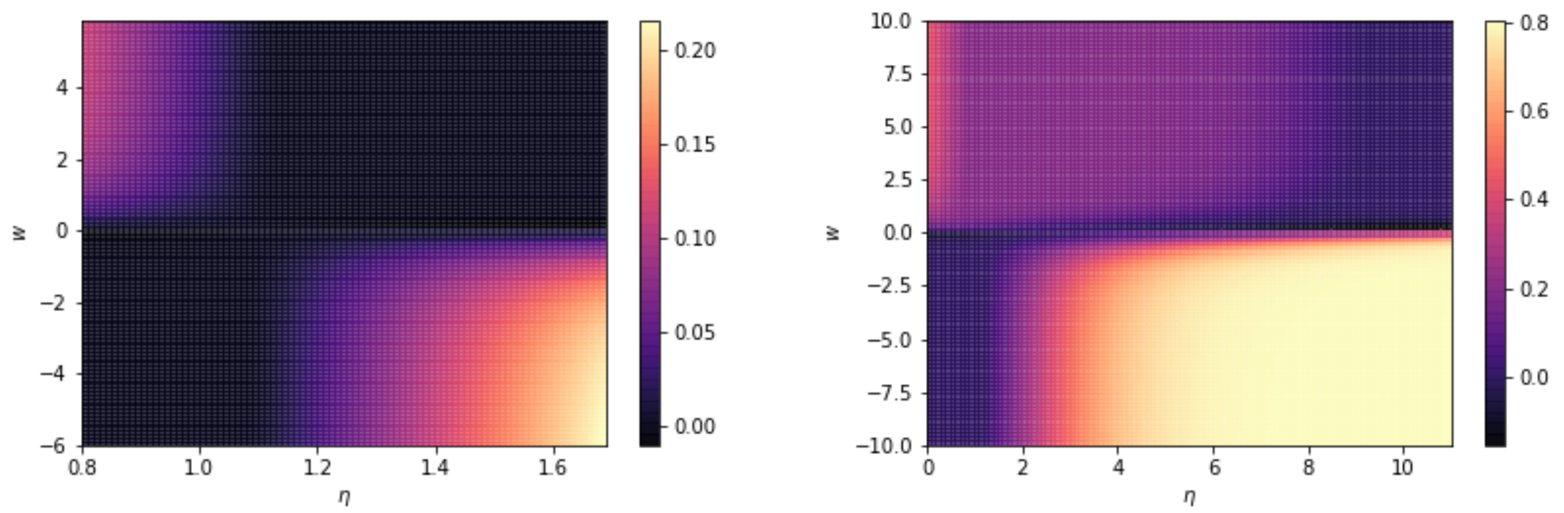


Figure: Heatmaps of the landscape of $F(\eta, w) = \sup_b [E_P \text{sigmoid}(wX + b) - E_{N(\eta,1)} \text{sigmoid}(wX + b)]$, where b is maximized out for visualization. Left: samples are drawn from $P = (1 - \epsilon)N(1, 1) + \epsilon N(1.5, 1)$ with $\epsilon = 0.2$. Right: samples are drawn from $P = (1 - \epsilon)N(1, 1) + \epsilon N(10, 1)$ 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 η does not consistently increase or decrease the value of η. 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 η, 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 η 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 η will only increase the value of η (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 η.

The Original JS-GAN

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

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

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

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

✗ ✓ ✓

JS-GAN

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

numerical experiment

$$X_1, ..., X_n \sim (1-\epsilon)N(\theta, I_p) + \epsilon N(\widetilde{\theta}, I_p)$$

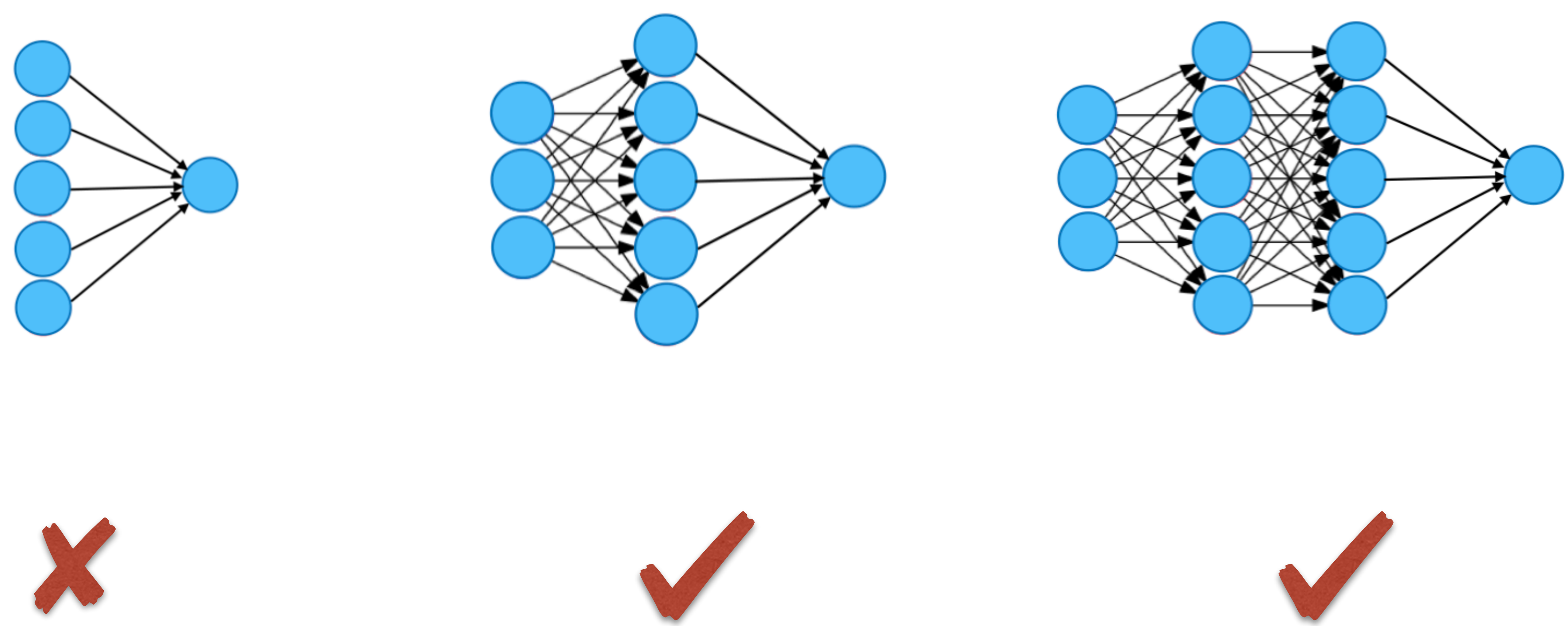
✗ ✓ ✓

JS-GAN

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

numerical experiment

$$X_1, ..., X_n \sim (1-\epsilon) N(\theta, I_p) + \epsilon N(\widetilde{\theta}, I_p)$$

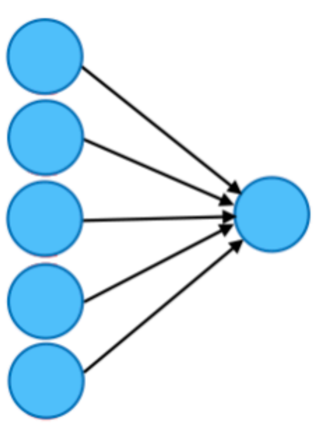


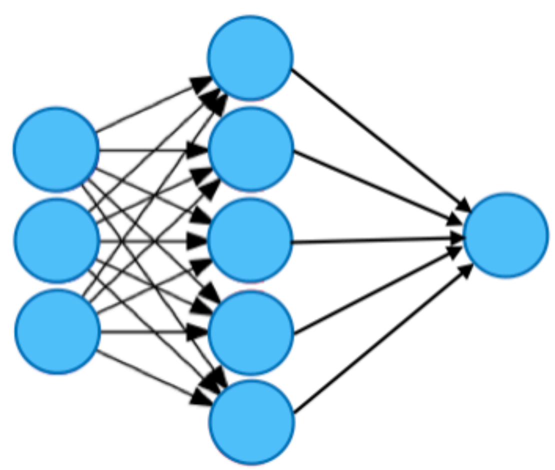
JS-GAN

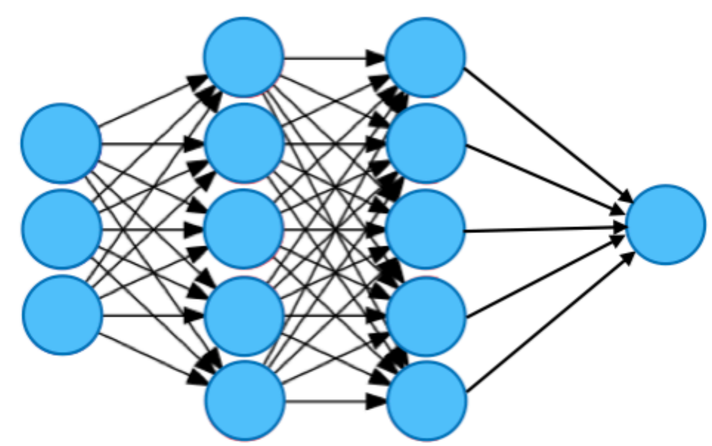
$$\widehat{\theta} = \underset{\eta \in \mathbb{R}^p}{\operatorname{argmin}} \max_{T \in \mathcal{T}} \left[\frac{1}{n} \sum_{i=1}^{n} \log T(X_i) + E_\eta \log(1 - T(X)) \right] + \log 4$$

numerical experiment

$$X_1, ..., X_n \sim (1-\epsilon) N(\theta, I_p) + \epsilon N(\widetilde{\theta}, I_p)$$







$$\widehat{\theta} \approx (1-\epsilon)\theta + \epsilon \widetilde{\theta}$$

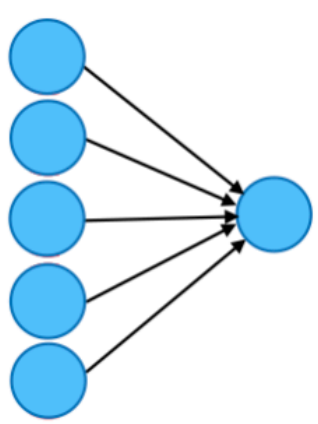
✗ ✓ ✓

JS-GAN

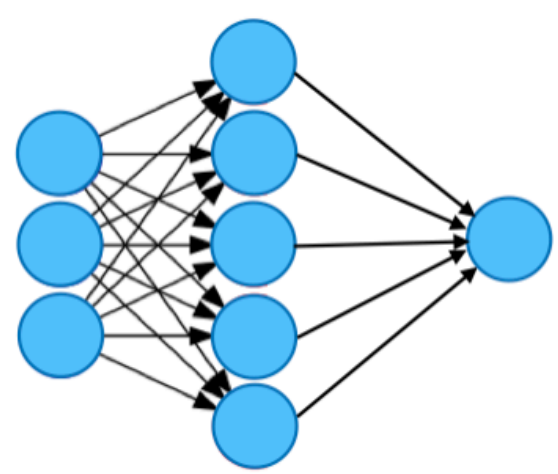
$$\widehat{\theta} = \underset{\eta \in \mathbb{R}^p}{\operatorname{argmin}} \max_{T \in \mathcal{T}} \left[\frac{1}{n} \sum_{i=1}^{n} \log T(X_i) + E_\eta \log(1 - T(X)) \right] + \log 4$$

numerical experiment

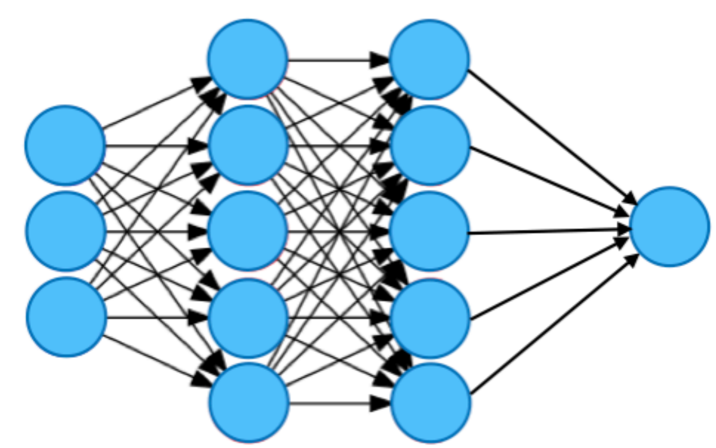
$$X_1, ..., X_n \sim (1-\epsilon)N(\theta, I_p) + \epsilon N(\widetilde{\theta}, I_p)$$



$$\widehat{\theta} \approx (1-\epsilon)\theta + \epsilon\widetilde{\theta}$$



$$\widehat{\theta} \approx \theta$$



$$\widehat{\theta} \approx \theta$$

JS-GAN

A classifier with hidden layers leads to robustness. Why?

JS-GAN

A classifier with hidden layers leads to robustness. Why?

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

JS-GAN

A classifier with hidden layers leads to robustness. Why?

$$\mathsf{JS}_g(\mathbb{P},\mathbb{Q}) = \max_{w\in\mathbb{R}^d}\left[\mathbb{P}\log\frac{1}{1+e^{-w^Tg(X)}} + \mathbb{Q}\log\frac{1}{1+e^{w^Tg(X)}}\right] + \log 4.$$

Proposition.

$$\mathsf{JS}_g(\mathbb{P},\mathbb{Q}) = 0 \iff \mathbb{P}g(X) = \mathbb{Q}g(X)$$

JS-GAN

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

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

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

with high probability uniformly over $\theta \in \mathbb{R}^p, Q$.

JS-GAN: Adaptation to Unknown Covariance

unknown covariance?

$$X_1, ..., X_n \sim (1-\epsilon)N(\theta, \Sigma) + \epsilon Q$$

JS-GAN: Adaptation to Unknown Covariance

unknown covariance?

$$X_1, ..., X_n \sim (1-\epsilon)N(\theta, \Sigma) + \epsilon Q$$

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

JS-GAN: Adaptation to Unknown Covariance

unknown covariance?

$$X_1, ..., X_n \sim (1-\epsilon)N(\theta, \Sigma) + \epsilon Q$$

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

no need to change the discriminator class

Generalization

Strong Contamination model:

$$X_1, ..., X_n \stackrel{iid}{\sim} P \quad \text{for some } P \text{ satisfying } \mathsf{TV}(P, E(\theta, \Sigma, H)) \leq \epsilon$$

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

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

$$\begin{cases} S(t, 1) = G(t) + (1 - t)G'(t), \\ S(t, 0) = G(t) - tG'(t). \end{cases} \tag{10}$$

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

$$\begin{aligned} \|\widehat{\theta} - \theta\|^2 &\leq C\left(\frac{p}{n} \vee \epsilon^2\right), \\ \|\widehat{\Sigma} - \Sigma\|_{\mathrm{op}}^2 &\leq C\left(\frac{p}{n} \vee \epsilon^2\right), \end{aligned}$$

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 $D_{\mathcal{T}}(P,Q)$, according to Proposition 3.1, is a variational lower bound of the Jensen-Shannon divergence

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

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 $D_{\mathcal{T}}(P,Q)$ is a variational lower bound of the total variation distance

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

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].

Example 4: Boosting Score

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 $D_{\mathcal{T}}(P,Q)$ 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

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

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

- *Dimension Halving:* [Lai et al.'16] `https://github.com/kal2000/AgnosticMeanAndCovarianceCode`.
- *Iterative Filtering:* [Diakonikolas et al.'17] `https://github.com/hoonose/robust-filter`.

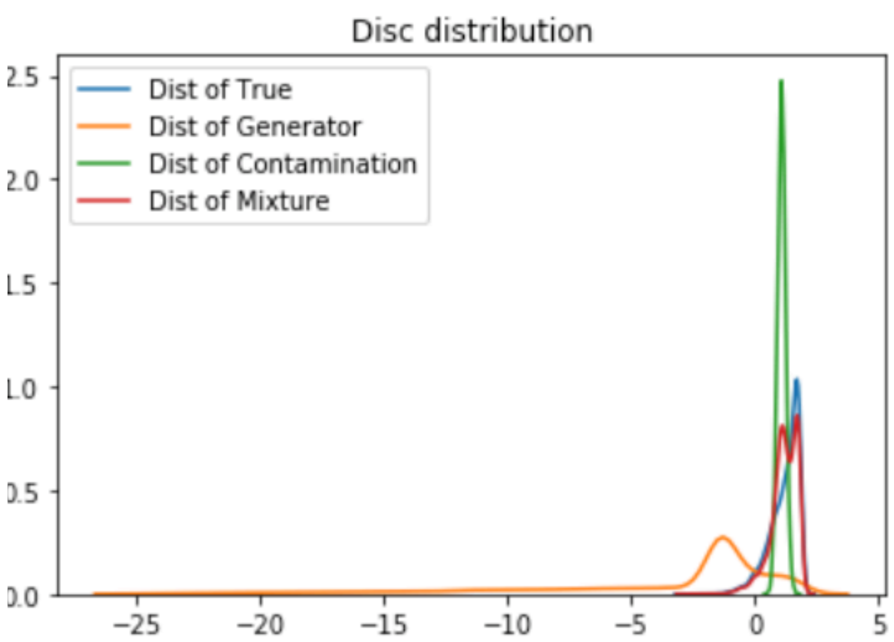
Robust Learning of Cauchy Distributions

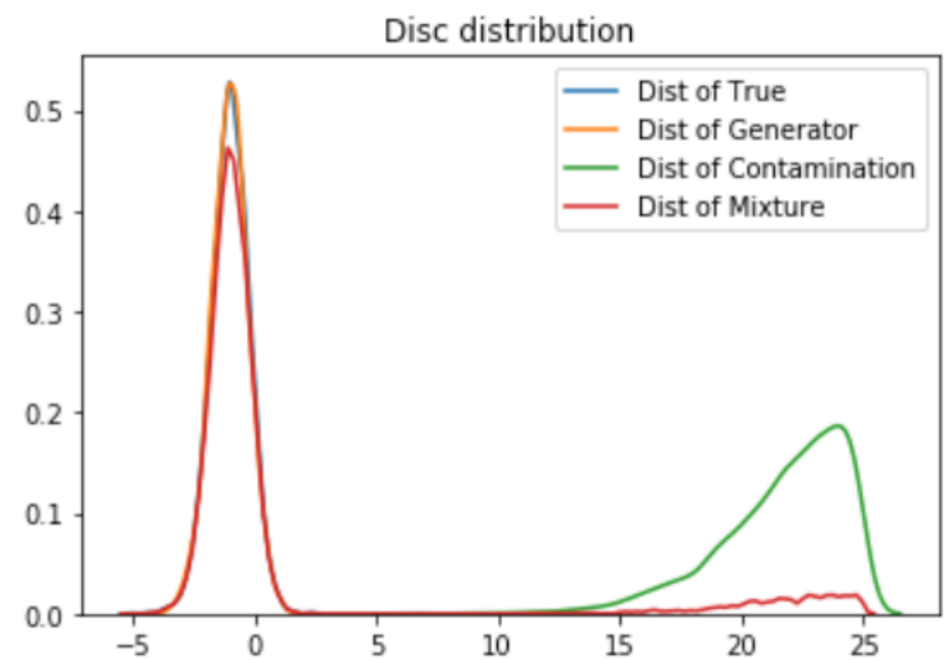
Table 4: Comparison of various methods of robust location estimation under Cauchy distributions. Samples are drawn from $(1-\epsilon)\text{Cauchy}(0_p, I_p) + \epsilon Q$ with $\epsilon = 0.2, p = 50$ and various choices of Q. Sample size: 50,000. Discriminator net structure: 50-50-25-1. Generator $g_\omega(\xi)$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

Contamination Q	JS-GAN (G_1)	JS-GAN (G_2)	Dimension Halving	Iterative Filtering
$\text{Cauchy}(1.5 * 1_p, I_p)$	**0.0664 (0.0065)**	0.0743 (0.0103)	0.3529 (0.0543)	0.1244 (0.0114)
$\text{Cauchy}(5.0 * 1_p, I_p)$	**0.0480 (0.0058)**	0.0540 (0.0064)	0.4855 (0.0616)	0.1687 (0.0310)
$\text{Cauchy}(1.5 * 1_p, 5 * I_p)$	0.0754 (0.0135)	**0.0742 (0.0111)**	0.3726 (0.0530)	0.1220 (0.0112)
$\text{Normal}(1.5 * 1_p, 5 * I_p)$	**0.0702 (0.0064)**	0.0713 (0.0088)	0.3915 (0.0232)	0.1048 (0.0288))

- *Dimension Halving*: [Lai et al.'16] `https://github.com/kal2000/AgnosticMeanAndCovarianceCode`.
- *Iterative Filtering*: [Diakonikolas et al.'17] `https://github.com/hoonose/robust-filter`.

Discriminator identifies outliers



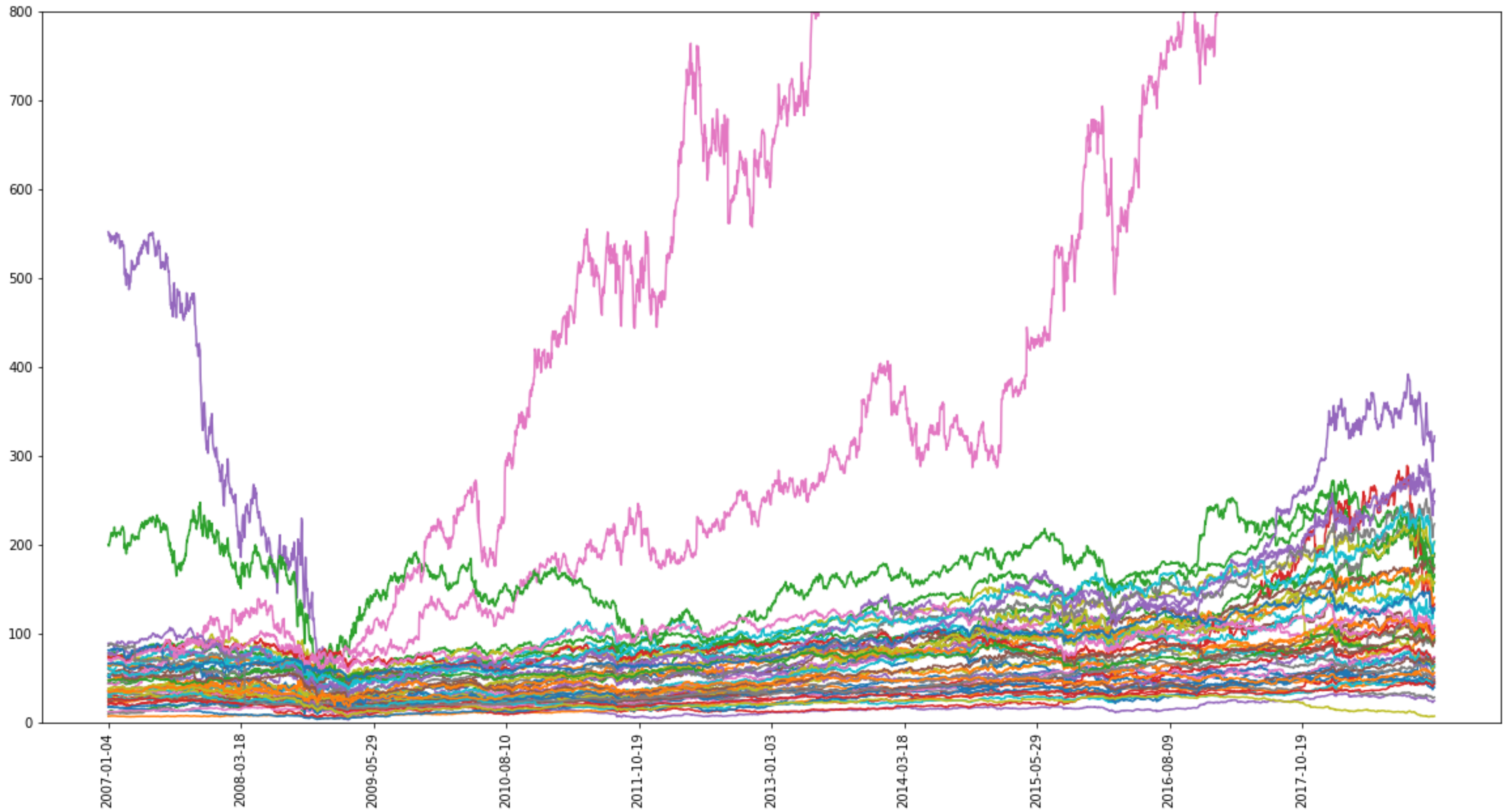


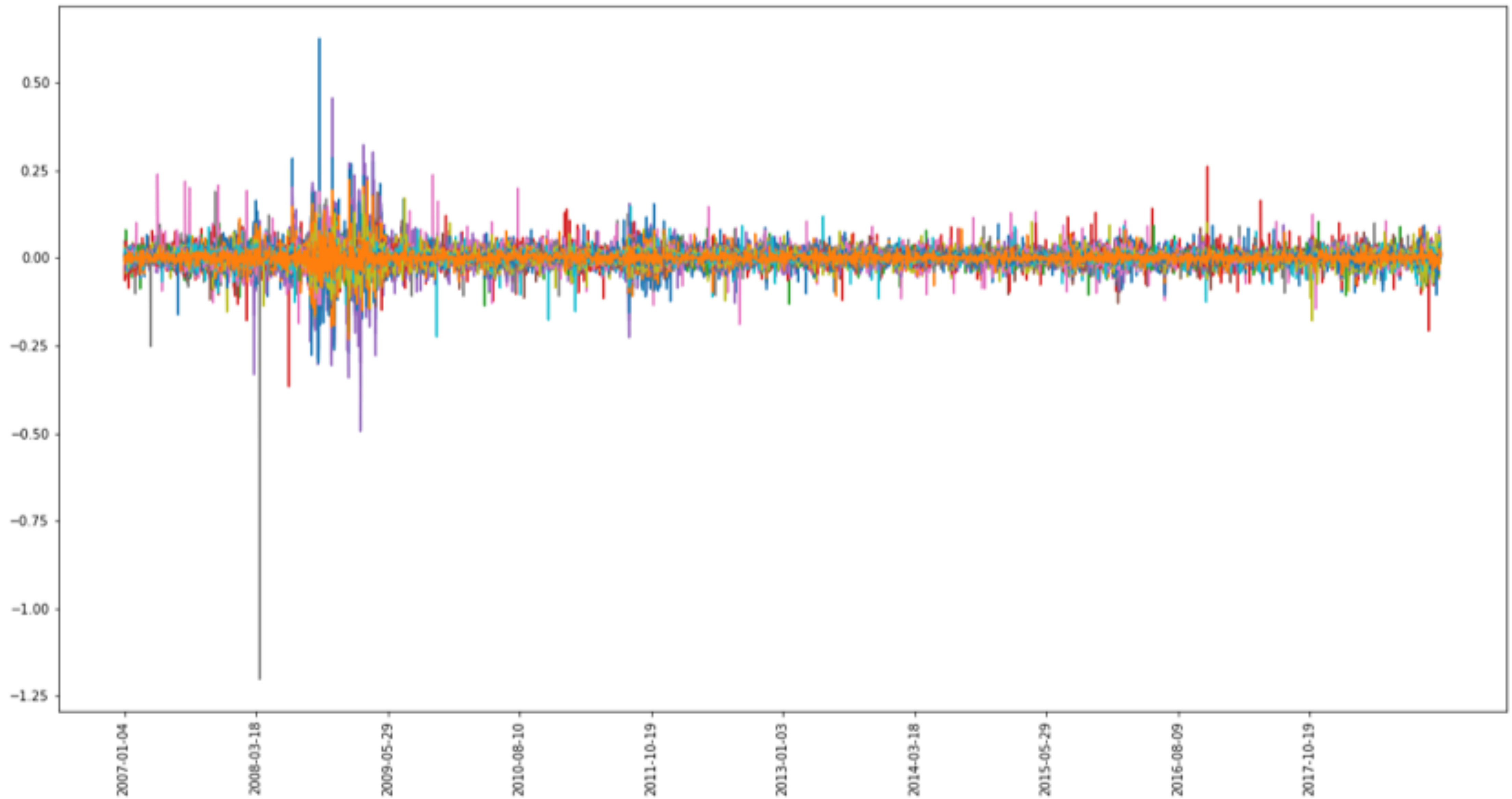
$$(1-\epsilon)N(0_p, I_p) + \epsilon Q$$

$$N(5 * 1_p, I_p)$$

- Discriminator helps identify outliers or contaminated samples
- Generator fits uncontaminated portion of true samples

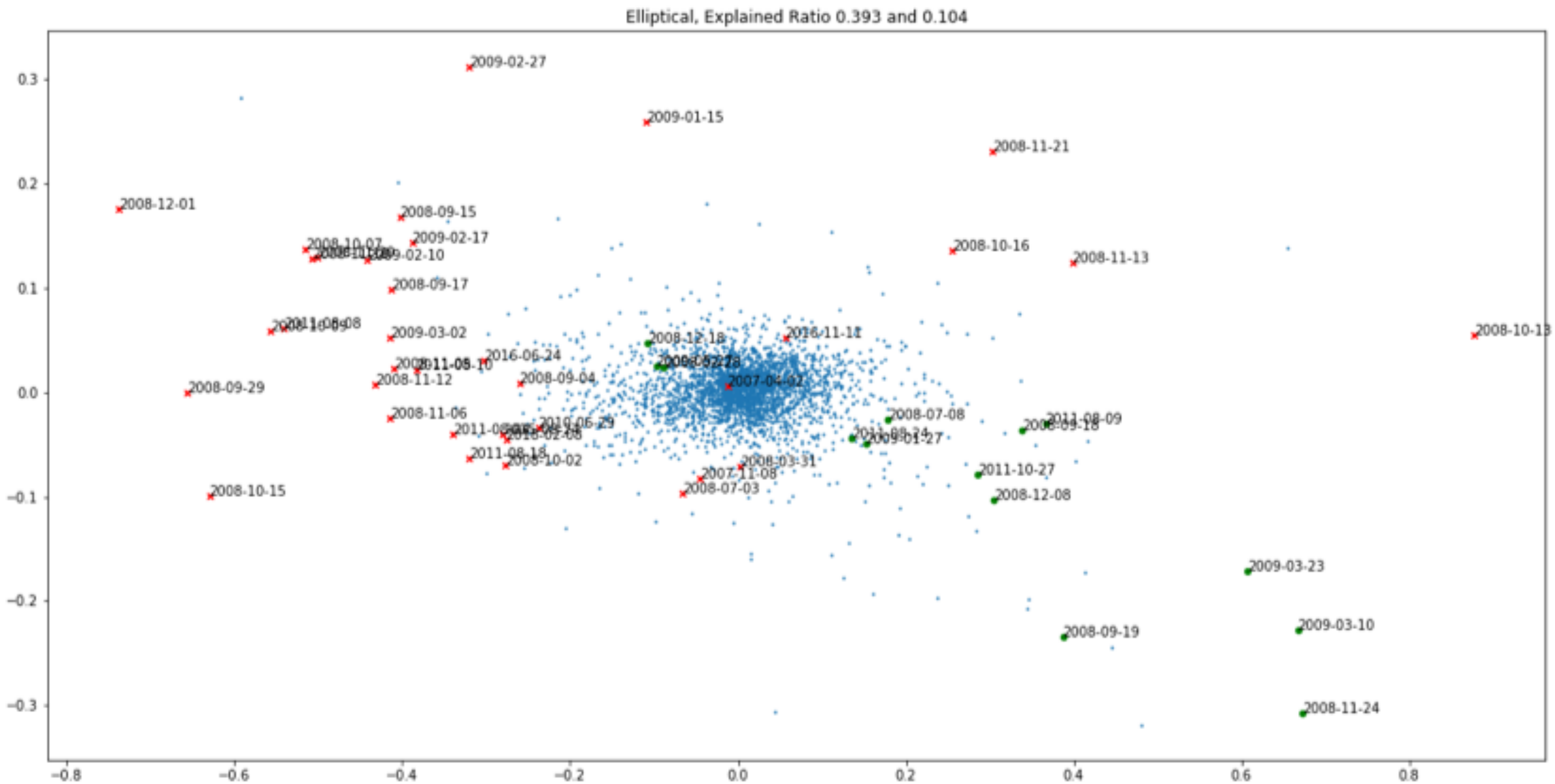
Application: Price of 50 stocks from 2007/01 to 2018/12
Corps are selected by ranking in market capitalization



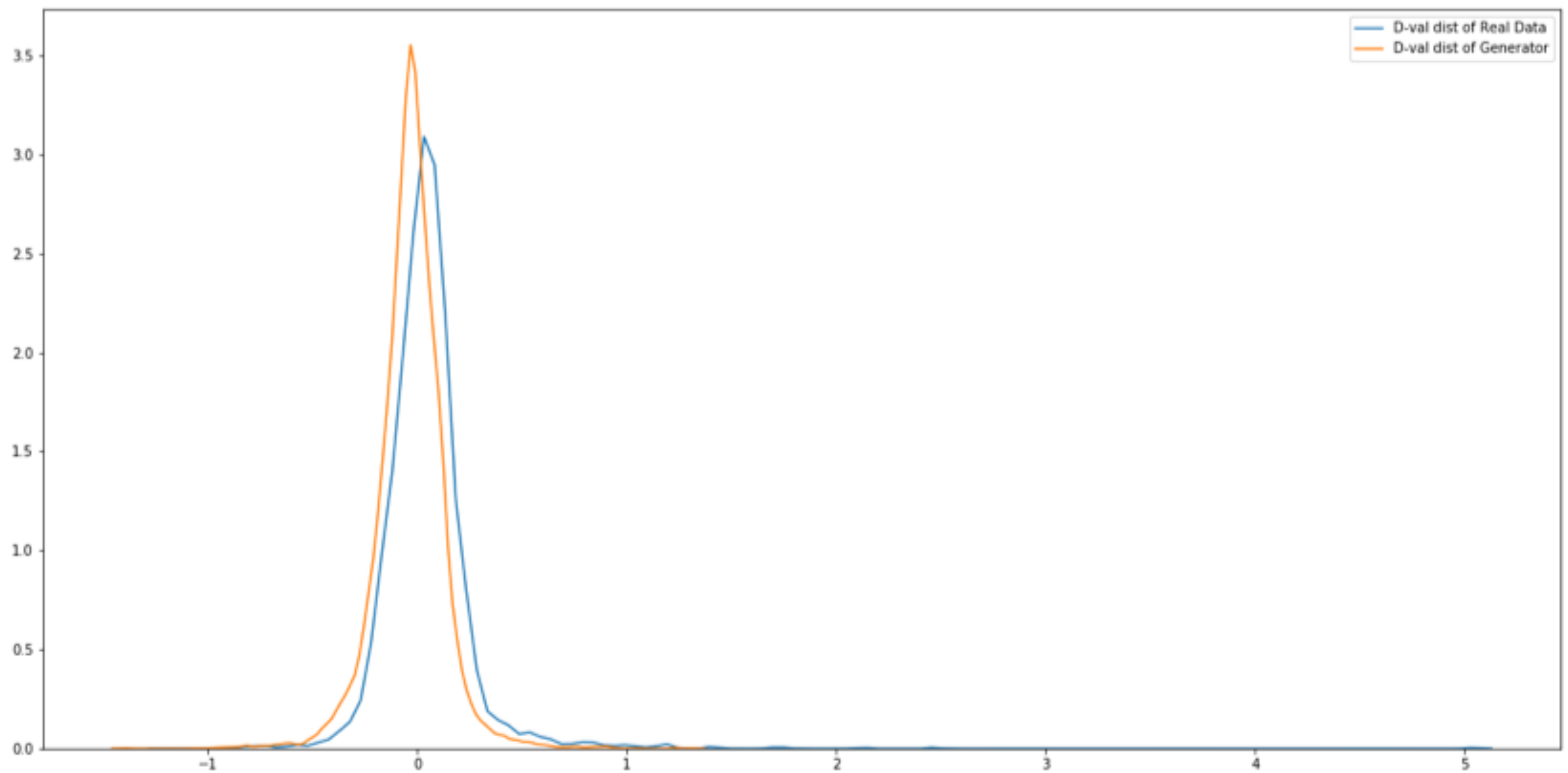


Log-return. y[i] = log(price_{i+1}/price_{i})

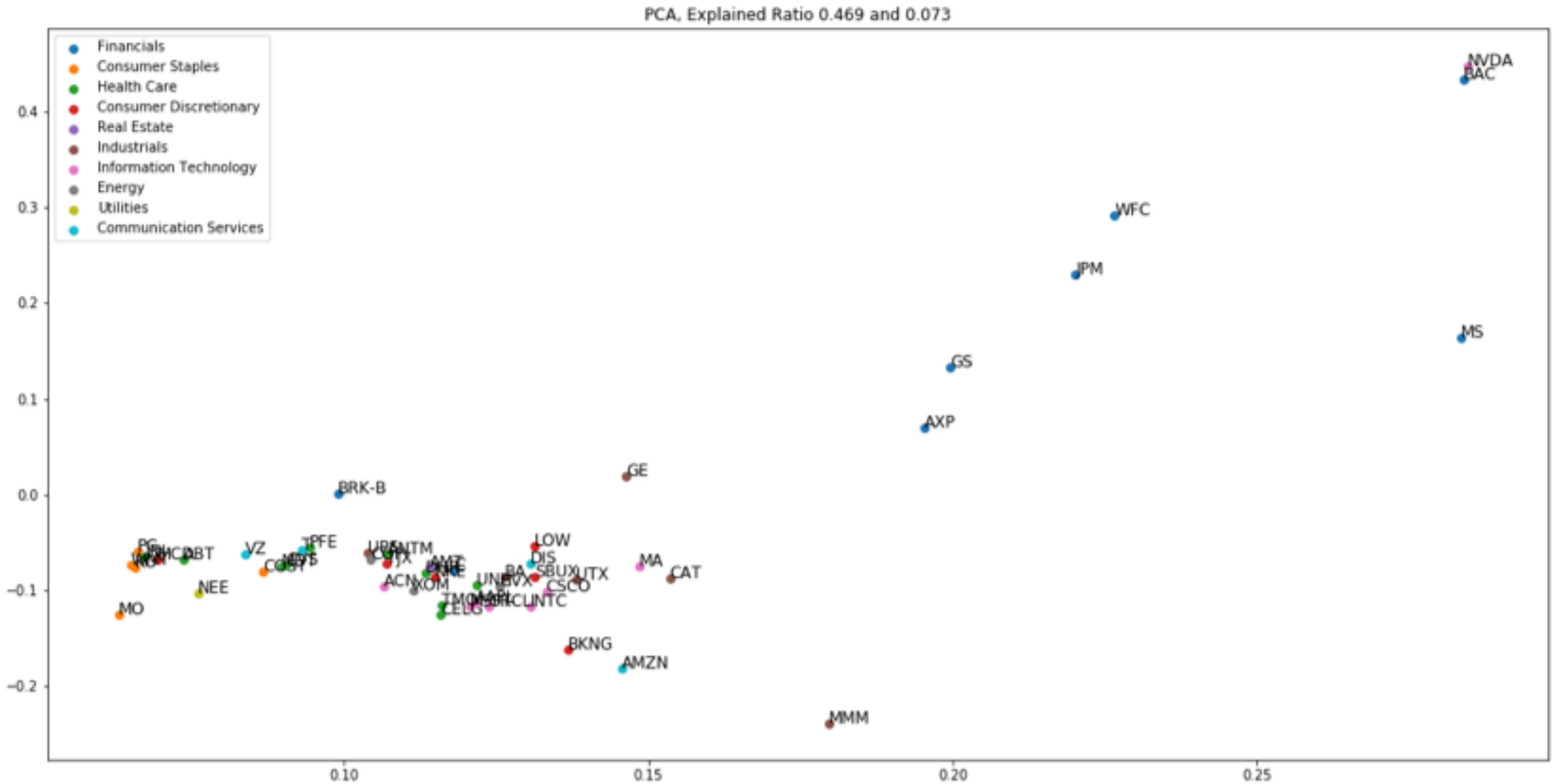
Fit data by Elliptica-GAN.
Apply SVD on scatter.
Dimension reduction on R^2.
outlier x and o are selected from Discriminator value distribution.



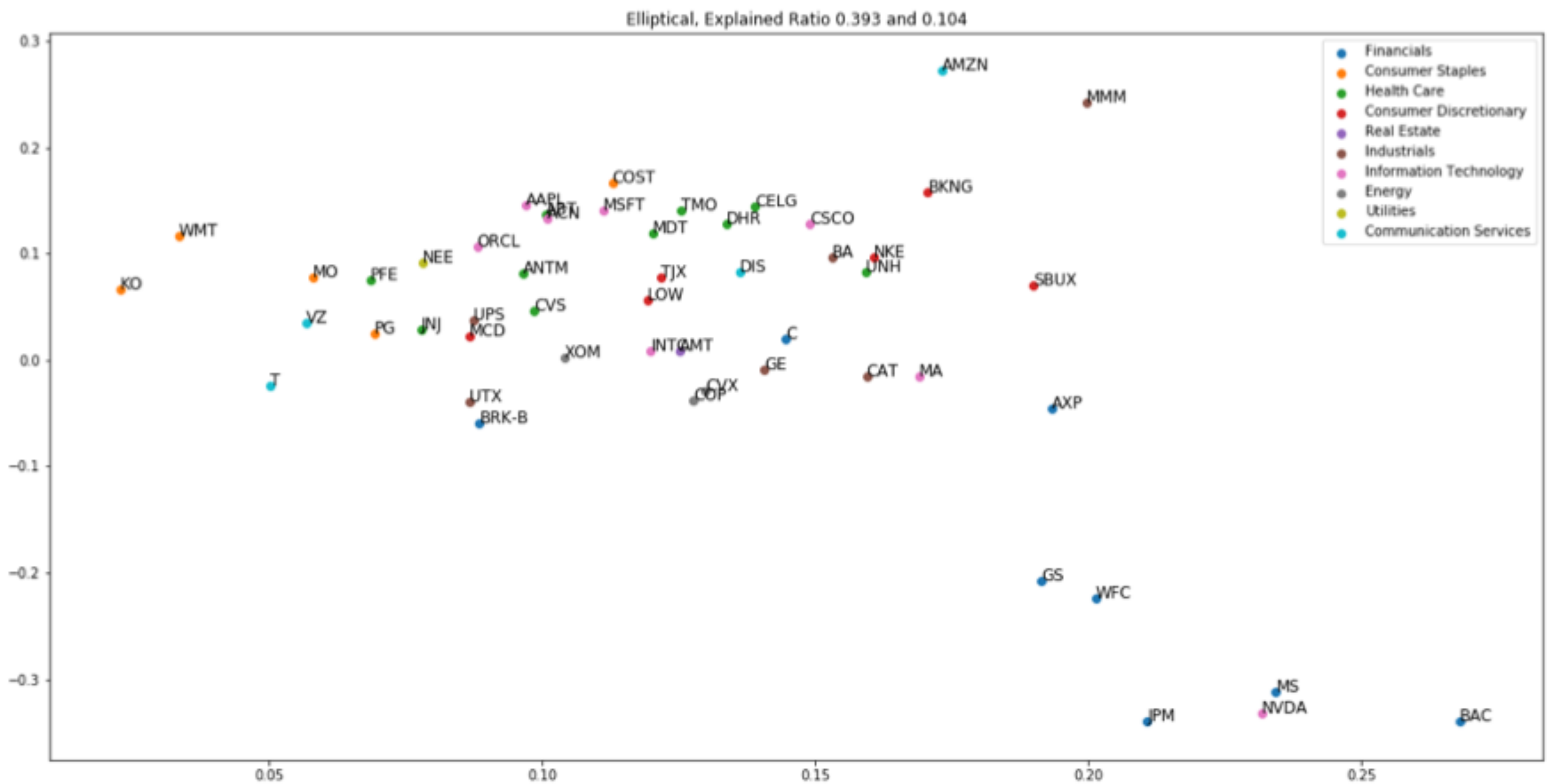
Discriminator value distribution from (Elliptical) Generator and real samples. Outliers are chosen from samples larger/ lower than a chosen percentile of Generator distribution



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

- Gao, Liu, Yao, Zhu, Robust Estimation and Generative Adversarial Networks, *ICLR 2019*, https://arxiv.org/abs/1810.02030

- Gao, Yao, Zhu, Generative Adversarial Networks for Robust Scatter Estimation: A Proper Scoring Rule Perspective, https://arxiv.org/abs/1903.01944

Thank You