MINE: Mutual Information Neural Estimation

ICML 2018

Motivation

- Mutual information was a powerful tool in statistical models:
  - Feature selection, information bottleneck, casualty
- MI quantifies the dependence of two random variables:

$$I(X;Z) = \int_{\mathcal{X}\times\mathcal{Z}} \log \frac{d\mathbb{P}_{XZ}}{d\mathbb{P}_X \otimes \mathbb{P}_Z} d\mathbb{P}_{XZ}$$

$$I(X;Z) := H(X) - H(X \mid Z)$$

Motivation

- MI is tractable only for discrete random variables or known probability distribution
- Common Approaches do not scale well with sample size or dimension:
  - Non-parametric approaches
  - Approximate gaussianity
- Use KL-Divergence for computing MI
- Use dual formulation for estimating f-divergence
  - Adversarial game between neural nets

MI

- KL-Divergence definition:

$$D_{KL}(\mathbb{P} \,||\, \mathbb{Q}) := \mathbb{E}_{\mathbb{P}}\left[\log \frac{d\mathbb{P}}{d\mathbb{Q}}\right]$$

- MI:

$$I(X;Z) = \int_{\mathcal{X}\times\mathcal{Z}} \log \frac{d\mathbb{P}_{XZ}}{d\mathbb{P}_X \otimes \mathbb{P}_Z} d\mathbb{P}_{XZ}$$

$$I(X,Z) = D_{KL}(\mathbb{P}_{XZ} \,||\, \mathbb{P}_X \otimes \mathbb{P}_Z)$$

MI Estimator

- Donsker-Varadhan representation:

$$D_{KL}(\mathbb{P} \,||\, \mathbb{Q}) = \sup_{T:\Omega \to \mathbb{R}} \mathbb{E}_{\mathbb{P}}[T] - \log(\mathbb{E}_{\mathbb{Q}}[e^{T}])$$

- So:

$$D_{KL}(\mathbb{P} \,||\, \mathbb{Q}) \geq \sup_{T \in \mathcal{F}} \mathbb{E}_{\mathbb{P}}[T] - \log(\mathbb{E}_{\mathbb{Q}}[e^{T}])$$

MI Estimator

- f-divergence representation:

$$D_{KL}(\mathbb{P} \,||\, \mathbb{Q}) \geq \sup_{T \in \mathcal{F}} \mathbb{E}_{\mathbb{P}}[T] - \mathbb{E}_{\mathbb{Q}}[e^{T-1}]$$

- Both representations are tight but Donsker-Varadhan representation is stronger as:

$$x \;\geq\; e \log x$$

  – Where:

$$\mathbb{E}_{\mathbb{Q}}[e^{T}]$$

Method

- Estimate function T using neural network:

$$I_\Theta(X,Z) = \sup_{\theta\in\Theta} \mathbb{E}_{\mathbb{P}_{XZ}}[T_\theta] - \log(\mathbb{E}_{\mathbb{P}_X \otimes \mathbb{P}_Z}[e^{T_\theta}])$$

- So:

$$I(X;Z) \geq I_\Theta(X,Z)$$

- Estimate $I_\Theta(X,Z)$ using:

$$\widehat{I(X;Z)}_n = \sup_{\theta\in\Theta} \mathbb{E}_{\mathbb{P}_{XZ}^{(n)}}[T_\theta] - \log(\mathbb{E}_{\mathbb{P}_X^{(n)} \otimes \hat{\mathbb{P}}_Z^{(n)}}[e^{T_\theta}])$$

Algorithm

Algorithm 1 MINE

$\theta \leftarrow$ initialize network parameters
repeat
Draw b minibatch samples from the joint distribution:
$(\boldsymbol{x}^{(1)}, \boldsymbol{z}^{(1)}), \ldots, (\boldsymbol{x}^{(b)}, \boldsymbol{z}^{(b)}) \sim \mathbb{P}_{XZ}$
Draw n samples from the Z marginal distribution:
$\bar{\boldsymbol{z}}^{(1)}, \ldots, \bar{\boldsymbol{z}}^{(b)} \sim \mathbb{P}_Z$
Evaluate the lower-bound:
$\mathcal{V}(\theta) \leftarrow \frac{1}{b}\sum_{i=1}^{b} T_\theta(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}) - \log(\frac{1}{b}\sum_{i=1}^{b} e^{T_\theta(\boldsymbol{x}^{(i)}, \bar{\boldsymbol{z}}^{(i)})})$
Evaluate bias corrected gradients (e.g., moving average):
$\widehat{G}(\theta) \leftarrow \widetilde{\nabla}_\theta \mathcal{V}(\theta)$
Update the statistics network parameters:
$\theta \leftarrow \theta + \widehat{G}(\theta)$
until convergence

Caveats

- Mini-batch computation is biased:

$$\widehat{G}_B = \mathbb{E}_B[\nabla_\theta T_\theta] - \frac{\mathbb{E}_B[\nabla_\theta T_\theta \, e^{T_\theta}]}{\mathbb{E}_B\left[e^{T_\theta}\right]}$$

  – Moving Average for estimating $\mathbb{E}_B\left[e^{T_\theta}\right]$ over full batch

- MI is not bounded and can become infinitely large so it will mask cross-entropy loss:

$$g_a = \min(||g_u|| \, , ||g_m||) \, \frac{g_m}{||g_m||}$$

Properties

- Strong consistency:

$$\forall n \geq N, \quad |I(X,Z) - \widehat{I(X;Z)}_n| \leq \epsilon$$

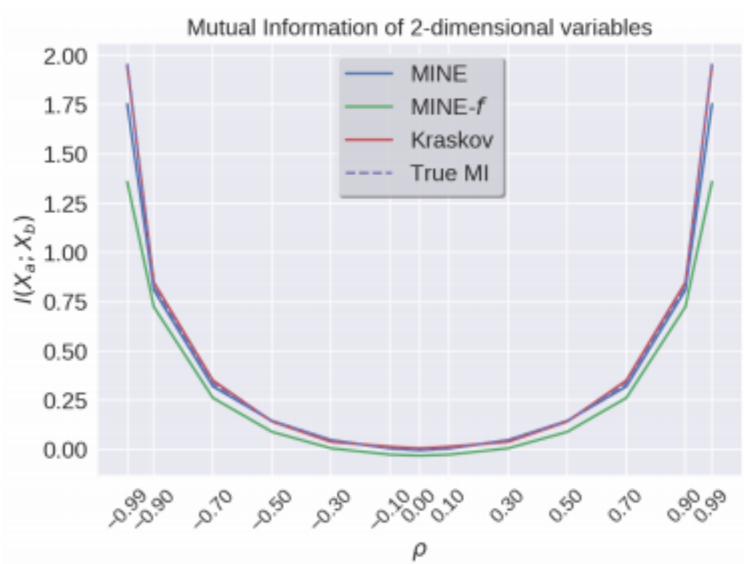
  - Lemma 1: $|I(X,Z) - I_\Theta(X,Z)| \leq \epsilon$
  - Lemma 2: $\forall n \geq N, \quad | \widehat{I(X;Z)}_n - I_\Theta(X,Z) | \leq \epsilon$

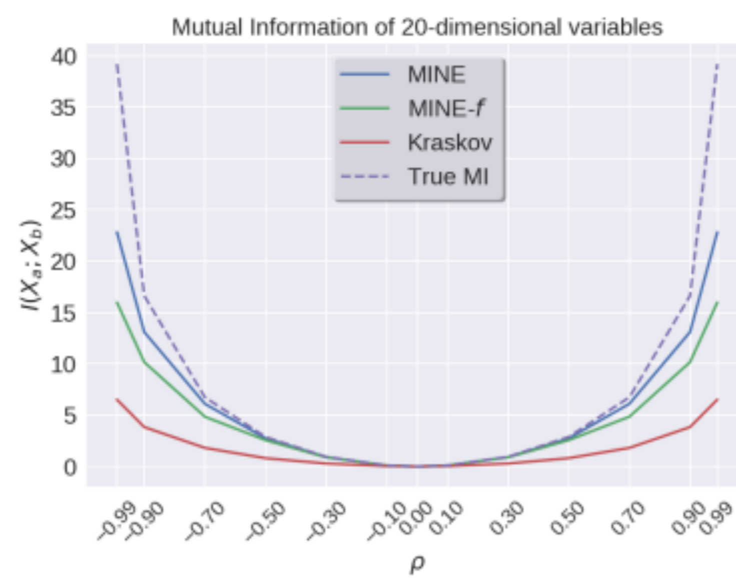
- Sample Complexity:

$$\widetilde{O}\left(\frac{d \log d}{\epsilon^2}\right)$$

Comparing to non-parametric estimator

- Two random variables with multivariate Gaussians distribution
- K-NN based estimator
- MINE and MINE-f

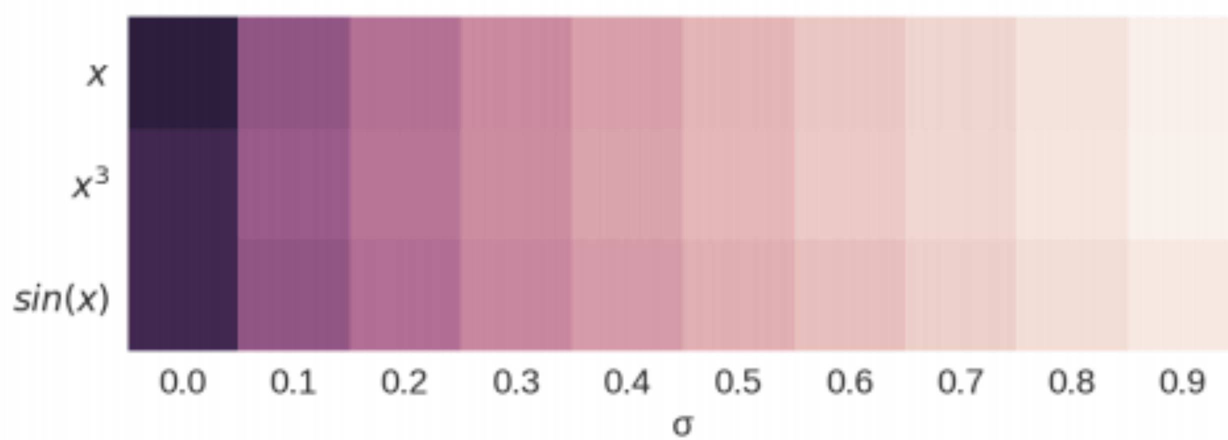




Capturing Non-Linear Dependency

- MI is a good measure for capturing non-linearity

$$Y = f(X) + \sigma \odot \epsilon$$



Improving GAN

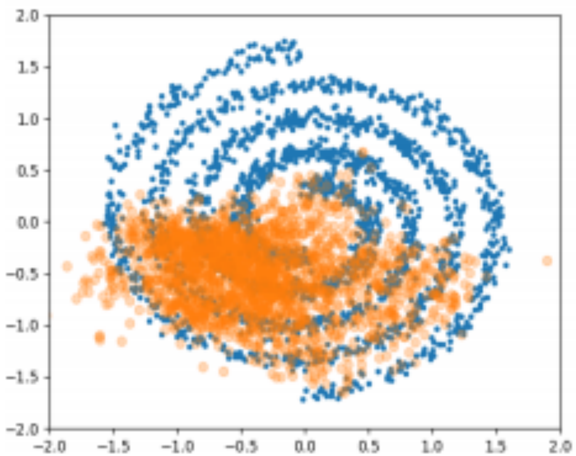
- GAN objective:

$$\min_G \max_D V(D,G) := \mathbb{E}_{\mathbb{P}_X}[D(X)] + \mathbb{E}_{\mathbb{P}_Z}[\log(1 - D(G(Z))]$$

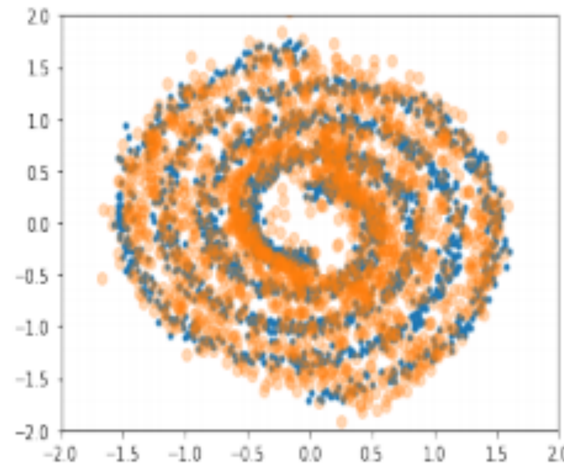
- Mode Collapse:
  - All generated samples are similar
- Maximize the MI between generated samples and code:

$$\arg\max_G \mathbb{E}[\log(D(G([\boldsymbol{\epsilon}, \boldsymbol{c}])))] + \beta I(G([\boldsymbol{\epsilon}, \boldsymbol{c}]); \boldsymbol{c})$$

Improving GAN - Result



(a) GAN



(b) GAN+MINE

	Stacked MNIST	
	Modes (Max 1000)	KL
DCGAN	99.0	3.40
ALI	16.0	5.40
Unrolled GAN	48.7	4.32
VEEGAN	150.0	2.95
PacGAN	1000.0 ± 0.0	$0.06 \pm 1.0\mathrm{e}^{-2}$
GAN+MINE (Ours)	1000.0 ± 0.0	$0.05 \pm 6.9\mathrm{e}^{-3}$

Bi-Directional Adversarial Model

- Encode input and reconstruct it from its encoding:
  - Encoder: $p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z} \mid \boldsymbol{x})p(\boldsymbol{x})$
  - Decoder: $q(\boldsymbol{x}, \boldsymbol{z}) = q(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z})$
- Reconstruction error:

$$\mathcal{R} \leq D_{KL}(q(\boldsymbol{x}, \boldsymbol{z}) \mid\mid p(\boldsymbol{x}, \boldsymbol{z})) - I_q(\boldsymbol{x}, \boldsymbol{z}) + H_q(\boldsymbol{z})$$

- Objectives:

$$\arg\max_{D} \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{z})}[\log D(\boldsymbol{x}, \boldsymbol{z})] + \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{z})}[\log(1 - D(\boldsymbol{x}, \boldsymbol{z}))]$$

$$\arg\max_{F,G} \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{z})}[\log(1 - D(\boldsymbol{x}, \boldsymbol{z}))] + \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{z})}[\log D(\boldsymbol{x}, \boldsymbol{z})] + \beta I_q(\boldsymbol{x}, \boldsymbol{z})$$

Bi-Directional Adversarial Models-Results

Model	Recons. Error	Recons. Acc.(%)	MS-SSIM
MNIST			
ALI	14.24	45.95	0.97
ALICE(l_2)	3.20	99.03	0.97
ALICE(Adv.)	5.20	98.17	0.98
MINE	9.73	96.10	0.99
CelebA			
ALI	53.75	57.49	0.81
ALICE(l_2)	8.01	32.22	0.93
ALICE(Adv.)	92.56	48.95	0.51
MINE	36.11	76.08	0.99

Information Bottleneck

- Find representation Z for X which has enough data for predating Y and discards irrelevant information in X

$$\mathcal{L}[q(Z \mid X)] = H(Y|Z) + \beta I(X, Z)$$

Model	Misclass. rate(%)
Baseline	1.38%
Dropout	1.34%
Confidence penalty	1.36%
Label Smoothing	1.40%
DVB	1.13%
DVB + Additive noise	1.06%
MINE(Gaussian) (ours)	1.11%
MINE(Propagated) (ours)	1.10%
MINE(Additive) (ours)	1.01%

Questions?

Thanks