MINE: Mutual Information Neural Estimation - A Comprehensive Exploration

Student: Li Ge
Instructor: Prof. Wu Youlong
EE142 Final Project, ShanghaiTech University

Research Status of MI

Importance of MI

In contrast to correlation, MI captures **non-linear** statistical dependencies between variables, and thus can act as a measure of true dependence.

Challenges

Hard to conduct exact computation.

Existing common approaches

Non-parametric (binning, likelihood ratio, KDE, approximate gaussianity), which is **not linearly related to the dimension of data and sample size**, leading to hard computation for high dimension data and thus **not general-purpose**. Recent work uses a dual formulation to estimate MI (through f-divergences ,including the KL-divergence) by adversarial deep neural networks.

Main Ideas of MINE

- Exploiting **dual optimization** to estimate divergences goes beyond the minimax objective as formalized in GANs, which is demonstrated in the paper.
- MINE is designed based on dual representations of the KL-divergence with this idea, so that the model can be used in mutual information estimation, maximization, and minimization and is not confined to situation that involves an adversarial game.

Contributions

1. Introduce MINE, which is scalable, flexible, and completely trainable via back-prop, as well as provide a thorough theoretical analysis.
2. Show that the utility of this estimator (based on dual optimization) **transcends the minimax objective** as formalized in GANs.
3. **GAN applications**: apply MINE to palliate mode-dropping in GANs and to improve reconstructions and inference in Adversarially Learned Inference on large scale datasets.
4. **Information Bottleneck application**: use MINE to apply the Information Bottleneck method in a continuous setting, and show that this approach **outperforms variational bottleneck methods.**

Theoretical Background

Mutual information

Equivalent to the Kullback-Leibler (KL-) divergence between the joint $\mathbb{P}_{XZ}$, and the product of the marginals $\mathbb{P}_X \otimes \mathbb{P}_Z$:

$$I(X,Z) = D_{KL}(\mathbb{P}_{XZ} \| \mathbb{P}_X \otimes \mathbb{P}_Z) \tag{1}$$

Intuition: the larger the divergence between the joint and the product of the marginals, the stronger the dependence between X and Z.

Dual representations of the KL-divergence

– Key technical, alternative way to estimate KL-divergence and then alternative way to estimate MI.

- The Donsker-Varadhan representation:

$$D_{KL}(\mathbb{P}\|\mathbb{Q}) = \sup_{T:\Omega\to\mathbb{R}} \mathbb{E}_{\mathbb{P}}[T] - \log(\mathbb{E}_{\mathbb{Q}}[e^T]) \tag{2}$$

where the supremum is taken over all functions T such that the two expectations are finite (integrability constraints).
For $\mathcal{F}$ be any class of functions $T : \Omega \to R$ satisfying the integrability constraints of the theorem:

$$D_{KL}(\mathbb{P}\|\mathbb{Q}) \geq \sup_{T\in\mathcal{F}} \mathbb{E}_{\mathbb{P}}[T] - \log(\mathbb{E}_{\mathbb{Q}}[e^T]) \tag{3}$$

- The f-divergence representation:

$$D_{KL}(\mathbb{P}\|\mathbb{Q}) \geq \sup_{T\in\mathcal{F}} \mathbb{E}_{\mathbb{P}}[T] - \mathbb{E}_{\mathbb{Q}}[e^{T-1}] \tag{4}$$

- **Comparison**: - Comparison: for a fixed T, the Donsker-Varadhan bound is stronger (larger). Since log function restrain the growth compared to linear function by applying $x > e\log(x)$.

Main Method

Idea

Choose $\mathcal{F}$ to be the family of functions $T_\theta : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ parametrized by a deep neural network with parameters . We call this network the statistics network. Thus we derive the relation:

$$I(X;Z) \geq I_\Theta(X;Z) \tag{5}$$

where $I_\Theta(X;Z)$ is the neural information measure defined by the Donsker-Varadhan representation as:

$$I_\Theta(X,Z) = \sup_{\theta\in\Theta} \mathbb{E}_{\mathbb{P}_{XZ}}[T_\theta] - \log(\mathbb{E}_{\mathbb{P}_X\otimes\mathbb{P}_Z}[e^{T_\theta}]) \tag{6}$$

- $\mathbb{P}_{XZ}$ - unknown, sampled from paired data.
- $\mathbb{P}_X \otimes \mathbb{P}_Z$ - unknown, sample from data, or by shuffling another sample in the joint distribution and then calculate it's marginal probability.

Define MINE

$$\widehat{I(X;Z)}_n = \sup_{\theta\in\Theta} \mathbb{E}_{\mathbb{P}^{(n)}_{XZ}}[T_\theta] - \log(\mathbb{E}_{\mathbb{P}^{(n)}_X\otimes\mathbb{P}^{(n)}_Z}[e^{T_\theta}]) \tag{7}$$

where $\mathcal{F} = \{T\}$ is the set of functions parametrized by a neural network, and $\hat{\mathbb{P}^{(n)}}$ is the empirical distribution associated to n i.i.d. samples given a distribution $\mathbb{P}$. Also sometimes we do not use log function directly in the loss function, since the gradient becomes extremely large when the marginal mean tends to 0. Alternatively we may use the reciprocal of EAM to approximate log function, meanwhile detach this term to avoid gradient calculation. However we still use log every time when calculating MI.

Correcting bias from the stochastic gradients

Problem: In a mini-batch setting, the SGD gradients of MINE are biased. (Possible) bias source:

- Nonlinear transformations (such as log) are sensitive to small sample noise.
- Mini-batch samples are insufficient to fully represent the true distribution.
- Other dependencies between samples introduce additional errors.

Solution in paper - exponential moving average (EMA): For extremely small or large errors (which we analyzed earlier regarding their effects on nonlinear functions), due to the weight $(1-\alpha)^k$, **long-term cumulative effects are exponentially diluted**, resulting in smoother gradient updates and more stable training.

I also came up with some new solutions, which will be demonstrated later.

Theoretical Properties

CONSISTENCY

– Divided into two problems:

- An approximation problem: related to the size of the function family $\mathcal{F}$: is $|\mathcal{F}|$ large enough to approximate optimal function T^*? - indeed! By **Universal Approximation Theorems for neural networks**. - Lemma 1 (approximation). (Also increase the hidden dimension of network to guarantee this!).
Lemma 1 (approximation). *Let $\epsilon > 0$. There exists a neural network parametrizing functions T_θ with parameters θ in some compact domain $\Theta \subset \mathbb{R}^k$, such that*

$$|I(X,Z) - I_\Theta(X,Z)| \leq \epsilon, \text{ a.e.}$$

- An estimation problem: is estimated MI calculated from prior distribution (derived from sample results of two probabilities) approximate true MI with increasing sample size? - indeed! By classical consistency theorems for extremum estimators. - Lemma 2 (estimation).
Lemma 2 (estimation). *Let $\epsilon > 0$. Given a family of neural network functions T_θ with parameters θ in some bounded domain $\Theta \subset \mathbb{R}^k$, there exists an $N \in \mathbb{N}$, such that*

$$\forall n \geq N, |\widehat{I(X,Z)} - I_\Theta(X,Z)| \leq \epsilon, \text{ a.e.}$$

- Combining these two lemmas with the triangular inequality, we have ***MINE is strongly consistent.***

SAMPLE COMPLEXITY

$\tilde{O}(\frac{d\log d}{\epsilon^2})$, *where d is the dimension of the parameter space.*

Empirical Comparisons

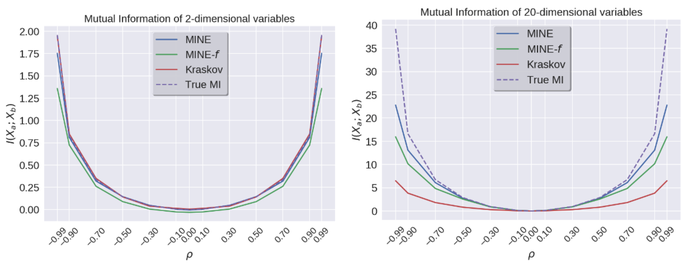
This part is aimed to show that MINE has good performance when estimating MI and accounts for non-linear dependence.

Comparing MINE to non-parametric estimation

- Between MINE, MINE-f and k-NN based non-parametric estimator.
- Consider multivariate Gaussian random variables: For multivariate Gaussian random variables X_a and X_b, when the marginal distributions are standardized Gaussian distributions, their MI can be conveniently calculated by the correlation coefficient ρ:

$$I(X_a;X_b) = -\frac{1}{2}\log(1-\rho^2), \text{ where } |\rho| < 1. \tag{8}$$

- Results:



Capturing non-linear dependencies

- **Equitability** - an important property of MI: For MI between random variables with relationship: $Y = f(X) + \sigma \odot \epsilon$, where f is a deterministic non-linear transformation, ϵ is random noise, and σ is a scalar representing the intensity of ϵ, their MI depends only on the noise, $\sigma \odot \epsilon$.
- Thus MINE is invariant to the choice of deterministic nonlinear transformation, consequently capturing non-linear statistical dependencies between variables.

Applications

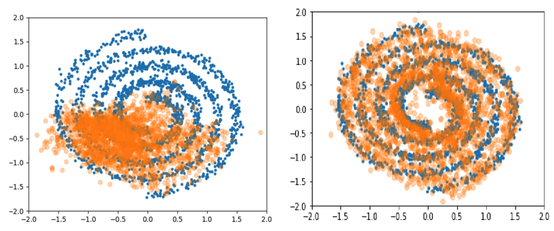
- **MI maximization**: to improve **mode representation** and **reconstruction** of generative models.
- **MI minimization**: to effectively implement the information bottleneck in a continuous setting.

Maximizing MI to improve GANs (mode coverage)

$$\arg\max_G E[log(D(G([\epsilon, c])))] + I(G([\epsilon, c]); c)$$

Network	Input	Input Meaning
G	z	Noise vector added to generator.
D	$(x, \tilde{x})$	True samples and generated samples.
MINE	$(z, \tilde{x})$	Noise vector and generated samples, in order to estimate $I(z;\tilde{x})$.

- Use adaptive gradient clipping: Since mutual information (MI) is theoretically unbounded, its gradient may differ significantly in magnitude compared to the gradient of the original loss component.
- Experiments: Spiral, 25-Gaussians datasets



(a) GAN (b) GAN+MINE

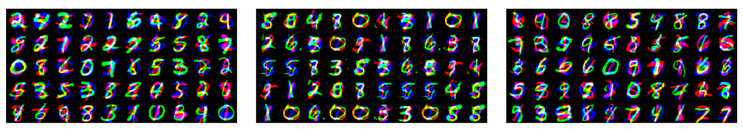
(a) Original data (b) GAN (c) GAN+MINE

The results demonstrate that the MINE-based regularization significantly improves mode coverage in GANs.

- Experiment: Stacked MNIST (with 1000 unique modes)

	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	$0.06 \pm 1.0e^{-2}$
GAN+MINE (Ours)	1000.0 ± 0.0	$0.05 \pm 6.9e^{-3}$

These results demonstrate that the MINE-based regularization enhances both the diversity (modes number) and accuracy (KL-divergence) of generated samples



(a) Training set (b) DCGAN (c) DCGAN+MINE

Maximizing MI to improve inference in bi-directional adversarial models (reconstruction)

- **Problem**: In practice ALI can lack fidelity (i.e., reconstructs less faithfully than desired), since for $X \to Z \to X'$, the training objective of ALI is to make the joint distributions $q(x,z)$ and $p(x,z)$ match as closely as possible, without optimizing reconstruction error (difference between X and X') directly and maximizing dependence between X' and Z:

$$\min_G \min_F \max_D \mathbb{E}_{p(x,z)}[\log D(x,z)] + \mathbb{E}_{q(x,z)}[\log(1 - D(x,z))]$$

- **Improvement**: Since the reconstruction error can be bounded by:

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

we can minimize the reconstruction error by maximizing $I_q(x,z)$. Thus we can modify the training objective as:

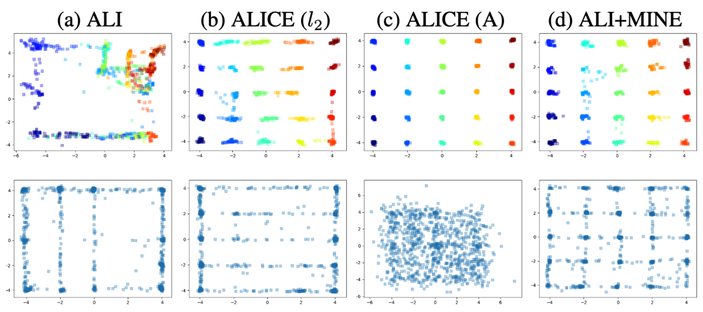
$$\arg\max_D \mathbb{E}_{q(x,z)}[\log D(x,z)] + \mathbb{E}_{p(x,z)}[\log(1 - D(x,z))]$$

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

- Experiment: ALI+MINE

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

ALICE has slightly better performance than MINE in reconstruction, while the problem of mode-dropping (loss reconstruction accuracy) is much worse.



Information Bottleneck

IB seeks an encoder, $q(Z \mid X)$, which includes Markovian structure $X \to Z \to Y$ by minimizing the loss function (IB Lagrangian):

$$\mathcal{L}[q(Z|X)] = H(Y|Z) + \beta I(X,Z) \tag{9}$$

- MINE can overcome the intractability of $I(X;Z)$ in the continuous setting of IB, and does not require a tractable density for the approximate posterior like variational bound methods.
- Related works includes discrete setting, jointly Gaussian continuous setting and variational bound.
- Experiment: Permutation-invariant MNIST classification
  - Define three types of encoders:
  1. A Gaussian encoder, same as DVB: $z = \mu(x) + \sigma \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$.
  2. An additive noise encoder: $z = \text{enc}(x + \sigma \odot \epsilon), \quad \epsilon \sim \mathcal{N}(0, I)$.
  3. A propagated noise encoder: $z = \text{enc}([x, \epsilon]), \quad \epsilon \sim \mathcal{N}(0, I)$.

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%

Conclusion

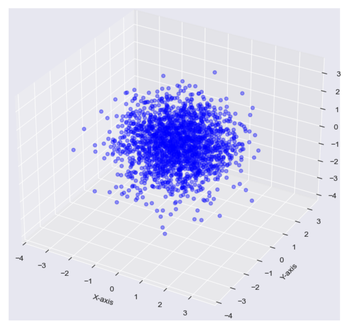
- A term of MI can be introduced alleviate mode-dropping issue in GANs.
- MI can also be used to improve inference and reconstructions in adversarially-learned inference.
- MI allows for tractable application of Information bottleneck methods in a continuous setting.

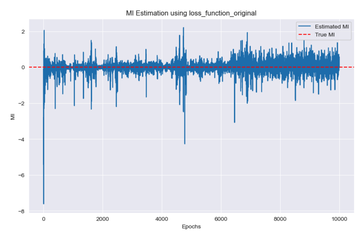
Reproduction of the Paper on 3-D Gaussian

Dataset: Multivariate Gaussian Distributions with standardized normal marginal, since their MI can be conveniently calculated by the correlation coefficient ρ.

Here I simply use the original loss function for MINE, which is (-MI):

$$\mathcal{L} = -\left(\mathbb{E}_{P(X,Z)}[t] - \log \mathbb{E}_{P(X)P(Z)}\left[e^t\right]\right)$$





Weakness - Numerical Instability: The term $\log \mathbb{E}[e^t]$ is highly sensitive to small variations in t, especially in small batch sizes. This can lead to large gradient fluctuations and potential instability during training.

Improved Loss Function

1. EAM loss function:

$$\mathcal{L} = -\left(\mathbb{E}_{P(X,Z)}[t] - \frac{1}{\mathbb{E}[\text{ma}(e^t)]} \cdot \mathbb{E}_{P(X)P(Z)}\left[e^t\right]\right)$$

Smoothing with Moving Average makes the optimization process smoother and reduces the sensitivity to the noise in the data.

2. Softplus loss function:

$$\mathcal{L} = -\left(\mathbb{E}_{P(X,Z)}[t] - \text{Softplus}\left(\mathbb{E}_{P(X)P(Z)}[e^t]\right)\right)$$

A smoothed approximation of the $\max(x, 0)$ function. It avoids extreme values for very large or very small t, improving numerical stability.

3. Clipped gradient loss function:

$$\mathcal{L} = -\left(\mathbb{E}_{P(X,Z)}[t] - \log \mathbb{E}_{P(X)P(Z)}[e^t]\right)$$

Weakness 1 - Gradient Distortion: Clipping modifies the true gradient direction, potentially leading to suboptimal convergence.
Weakness 2 - Ineffectiveness for Directional Instability: While clipping addresses magnitude issues, it does not resolve directional instability caused by noisy or poorly scaled gradients.

4. Log-sum-exp loss function:

$$\mathcal{L} = -\left(\mathbb{E}_{P(X,Z)}[t] - \log \sum_i e^{t_i - \max(t)}\right)$$

A standard technique for smoothing the gradient. **Specially**, it prevents numerical overflow or underflow, especially when computing sums of exponentials. **Also**, it constraints the sparse (like extremely small) data while still considering their contributions by considering the maximum.

5. Regularized loss function:

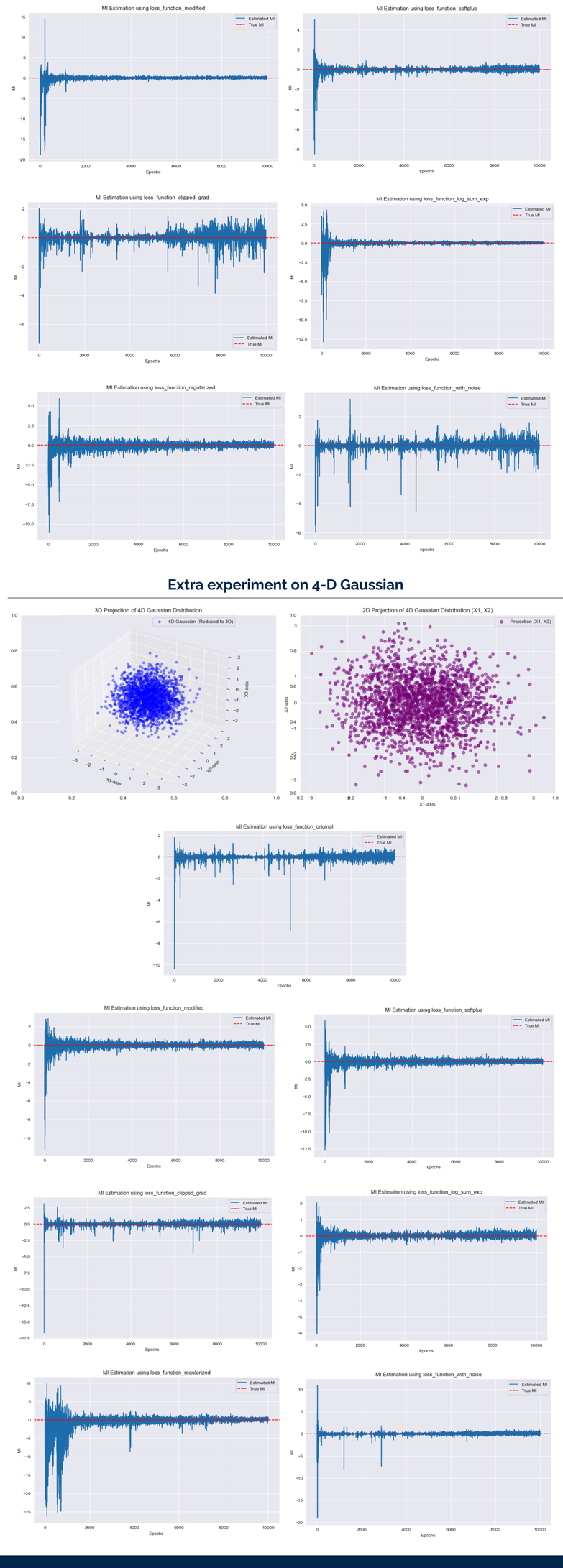
$$\mathcal{L} = -\left(\mathbb{E}_{P(X,Z)}[t] - \log \mathbb{E}_{P(X)P(Z)}[e^t]\right) + \lambda\|e^t\|_2^2$$

Besides smoothing the gradient like other modified loss, it also prevents overfitting and thus improves the model's generalization.

6. Noise loss function:

$$\mathcal{L} = -\left(\mathbb{E}_{P(X,Z)}[t + \mathcal{N}(0,\sigma^2)] - \log \mathbb{E}_{P(X)P(Z)}[e^t]\right)$$

Why add this?
1. To improve the model's robustness by encouraging it to learn more generalized representations.
2. To encourage the model to explore different directions during optimization.
Maybe the noise should be carefully chosen and combined with clipped gradient.



Extra experiment on 4-D Gaussian

lige2022@shanghaitech.edu.cn