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:

\[
I(X; Z) = \int_{X \times Z} \log \frac{dP_{XZ}}{dP_X \otimes P_Z} dP_{XZ}.
\]

\[
I(X; Z) := H(X) - H(X \mid Z)
\]
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:**
  \[
  D_{KL}(P \parallel Q) := E_{P} \left[ \log \frac{dP}{dQ} \right]
  \]

- **MI:**
  \[
  I(X; Z) = \int_{X \times Z} \log \frac{dP_{XZ}}{dP_X \otimes P_Z} dP_{XZ}
  \]
  \[
  I(X, Z) = D_{KL}(P_{XZ} \parallel P_X \otimes P_Z)
  \]
MI Estimator

• Donsker-Varadhan representation:

\[ D_{KL}(P || Q) = \sup_{T: \Omega \rightarrow \mathbb{R}} \mathbb{E}_P[T] - \log(\mathbb{E}_Q[e^T]) \]

• So:

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

$$I_{\Theta}(X, Z) = \sup_{\theta \in \Theta} E_{P_{XZ}}[T_{\theta}] - \log(E_{P_X \otimes P_Z}[e^{T_{\theta}}])$$

• So:

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

• Estimate $I_{\Theta}(X, Z)$ using:

$$\widehat{I(X; Z)}_n = \sup_{\theta \in \Theta} E_{P^{(n)}_{XZ}}[T_{\theta}] - \log(E_{P^{(n)}_X \otimes \hat{P}^{(n)}_Z}[e^{T_{\theta}}])$$
Algorithm 1 MINE

\[ \theta \leftarrow \text{initialize network parameters} \]

repeat

Draw \( b \) minibatch samples from the joint distribution:
\[
(\mathbf{x}^{(1)}, \mathbf{z}^{(1)}), \ldots, (\mathbf{x}^{(b)}, \mathbf{z}^{(b)}) \sim \mathbb{P}_{X,Z}
\]

Draw \( n \) samples from the \( Z \) marginal distribution:
\[
\tilde{z}^{(1)}, \ldots, \tilde{z}^{(b)} \sim \mathbb{P}_Z
\]

Evaluate the lower-bound:
\[
\mathcal{V}(\theta) \leftarrow \frac{1}{b} \sum_{i=1}^{b} T_{\theta}(\mathbf{x}^{(i)}, \mathbf{z}^{(i)}) - \log\left( \frac{1}{b} \sum_{i=1}^{b} e^{T_{\theta}(\mathbf{x}^{(i)}, \tilde{z}^{(i)})} \right)
\]

Evaluate bias corrected gradients (e.g., moving average):
\[
\hat{G}(\theta) \leftarrow \tilde{\nabla}_{\theta} \mathcal{V}(\theta)
\]

Update the statistics network parameters:
\[
\theta \leftarrow \theta + \hat{G}(\theta)
\]

until convergence
Caveats

• Mini-batch computation is biased:

\[
\hat{G}_B = \mathbb{E}_B[\nabla_\theta T_\theta] - \frac{\mathbb{E}_B[\nabla_\theta T_\theta e^{T_\theta}]}{\mathbb{E}_B[e^{T_\theta}]}
\]

– Moving Average for estimating \( \mathbb{E}_B[e^{T_\theta}] \) 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:

\[ \forall n \geq N, \quad |I(X, Z) - \hat{I}(X; Z)_n| \leq \epsilon. \]

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

• Sample Complexity:

\[ \tilde{O}\left(\frac{d \log d}{\epsilon^2}\right) \]
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:

\[
\min_G \max_D V(D, G) := E_{P_X}[D(X)] + E_{P_Z}[\log(1 - D(G(Z)))]
\]

• Mode Collapse:
  – All generated samples are similar

• Maximize the MI between generated samples and code:

\[
\arg\max_G \mathbb{E}[\log(D(G([\epsilon, c])))] + \beta I(G([\epsilon, c]; c))
\]
Improving GAN - Result

(a) GAN  
(b) GAN+MINE

<table>
<thead>
<tr>
<th>Stacked MNIST</th>
<th>Modes (Max 1000)</th>
<th>KL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCGAN</td>
<td>99.0</td>
<td>3.40</td>
</tr>
<tr>
<td>ALI</td>
<td>16.0</td>
<td>5.40</td>
</tr>
<tr>
<td>Unrolled GAN</td>
<td>48.7</td>
<td>4.32</td>
</tr>
<tr>
<td>VEEGAN</td>
<td>150.0</td>
<td>2.95</td>
</tr>
<tr>
<td>PacGAN</td>
<td>1000.0 ± 0.0</td>
<td>0.06 ± 1.0e^{-2}</td>
</tr>
<tr>
<td>GAN+MINE (Ours)</td>
<td>1000.0 ± 0.0</td>
<td>0.05 ± 6.9e^{-3}</td>
</tr>
</tbody>
</table>
Bi-Directional Adversarial Model

• Encode input and reconstruct it from its encoding:
  – Encoder:
    \[ p(x, z) = p(z \mid x)p(x) \]
  – Decoder:
    \[ q(x, z) = q(x \mid z)p(z) \]

• Reconstruction error:
  \[ R \leq D_{KL}(q(x, z) \mid\mid p(x, z)) - I_q(x, z) + H_q(z) \]

• Objectives:
  \[
  \begin{align*}
  \arg \max_{D} & \mathbb{E}_{q(x, z)}[\log D(x, z)] + \mathbb{E}_{p(x, z)}[\log (1 - D(x, z))] \\
  \arg \max_{F, G} & \mathbb{E}_{q(x, z)}[\log (1 - D(x, z))] + \mathbb{E}_{p(x, z)}[\log D(x, z)] + \beta I_q(x, z)
  \end{align*}
  \]
Bi-Directional Adversarial Models - Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Recons. Error</th>
<th>Recons. Acc. (%)</th>
<th>MS-SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MNIST</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALI</td>
<td>14.24</td>
<td>45.95</td>
<td>0.97</td>
</tr>
<tr>
<td>ALICE($l_2$)</td>
<td>3.20</td>
<td>99.03</td>
<td>0.97</td>
</tr>
<tr>
<td>ALICE(Adv.)</td>
<td>5.20</td>
<td>98.17</td>
<td>0.98</td>
</tr>
<tr>
<td>MINE</td>
<td>9.73</td>
<td>96.10</td>
<td>0.99</td>
</tr>
<tr>
<td><strong>CelebA</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALI</td>
<td>53.75</td>
<td>57.49</td>
<td>0.81</td>
</tr>
<tr>
<td>ALICE($l_2$)</td>
<td>8.01</td>
<td>32.22</td>
<td>0.93</td>
</tr>
<tr>
<td>ALICE(Adv.)</td>
<td>92.56</td>
<td>48.95</td>
<td>0.51</td>
</tr>
<tr>
<td>MINE</td>
<td>36.11</td>
<td>76.08</td>
<td>0.99</td>
</tr>
</tbody>
</table>
Information Bottleneck

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

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Misclass. rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>1.38%</td>
</tr>
<tr>
<td>Dropout</td>
<td>1.34%</td>
</tr>
<tr>
<td>Confidence penalty</td>
<td>1.36%</td>
</tr>
<tr>
<td>Label Smoothing</td>
<td>1.40%</td>
</tr>
<tr>
<td>DVB</td>
<td>1.13%</td>
</tr>
<tr>
<td>DVB + Additive noise</td>
<td>1.06%</td>
</tr>
<tr>
<td>MINE(Gaussian) (ours)</td>
<td>1.11%</td>
</tr>
<tr>
<td>MINE(Propagated) (ours)</td>
<td>1.10%</td>
</tr>
<tr>
<td>MINE(Additive) (ours)</td>
<td>1.01%</td>
</tr>
</tbody>
</table>
Questions?

Thanks