DEEP LEARNING JP
[DL Papers]

GANとエネルギーベースモデル

Shohei Taniguchi, Matsuo Lab (M2)

概要

以下の3本の論文をベースに、GANとEBMの関係についてまとめる

Deep Directed Generative Models with Energy-Based Probability Estimation
https://arxiv.org/abs/1606.03439

Maximum Entropy Generators for Energy-Based Models
https://arxiv.org/abs/1901.08508

Your GAN is Secretly an Energy-based Model and You Should use Discriminator Driven Latent Sampling
https://arxiv.org/abs/2003.06060

Outline

前提知識

- Generative Adversarial Network
- Energy-based Model

GANとEBMの類似点

論文紹介

Generative Adversarial Network
[Goodfellow et al., 2014]

識別器 D_θ と生成器 G_ϕ のミニマックスゲーム

$D_\theta : \mathbb{R}^{d_x} \to [0,1], G_\phi : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$

$$\mathscr{L}\left(\theta, \phi\right) = \mathbb{E}_{p(x)}\left[\log D_\theta(x)\right] + \mathbb{E}_{p(z)}\left[\log\left(1 - D_\theta\left(G_\phi(z)\right)\right)\right]$$

識別器は $\mathscr{L}$ を最大化し、生成器は $\mathscr{L}$ を最小化

GANの学習

GANの更新式は

$$\mathscr{L}\left(\theta,\phi\right)=\sum_{i=1}^{N}\log D_\theta(x_i)+\log\left(1-D_\theta\left(G_\phi\left(z_i\right)\right)\right)$$

$$\theta\leftarrow\theta+\eta_\theta\nabla_\theta\mathscr{L}\left(\theta,\phi\right)$$

$$\phi\leftarrow\phi-\eta_\phi\nabla_\phi\mathscr{L}\left(\theta,\phi\right)$$

$$z_i\sim\mathrm{Normal}\,(0,I)$$

GANの一般的な解釈
識別器＝密度比推定器

識別器はデータ分布 $p(x)$ と生成サンプルの分布 $p_\phi(x) = \mathbb{E}_{p(z)}\left[p\left(G_\phi(z)\right)\right]$ の密度比推定器としての役割を果たす

i.e., 識別器が最適なとき

$$D_\theta^*(x) = \frac{p(x)}{p(x) + p_\phi(x)}$$

GANの一般的な解釈
生成器の学習はJS divergenceの最小化

識別器が最適なとき

$$\mathscr{L}\left(\theta, \phi\right) = \mathrm{JS}\left(p\left(x\right) \parallel p_{\phi}\left(x\right)\right) - 2\log 2$$

$$\mathrm{JS}\left(p\left(x\right) \parallel p_{\phi}\left(x\right)\right) = \frac{1}{2}\mathrm{KL}\left(p\left(x\right) \parallel \frac{p\left(x\right) + p_{\phi}\left(x\right)}{2}\right) + \frac{1}{2}\mathrm{KL}\left(p_{\phi}\left(x\right) \parallel \frac{p\left(x\right) + p_{\phi}\left(x\right)}{2}\right)$$

生成器 G_{ϕ} はデータ分布とのJensen-Shannon divergence最小化により学習される

$-\log D$ Trick

オリジナルのロスだと、勾配消失が起こりやすいので、
後半を以下のように置き換えるトリックがよく使われる

$$\mathscr{L}\left(\theta, \phi\right) = \mathbb{E}_{p(x)}\left[\log D_\theta(x)\right] - \mathbb{E}_{p(z)}\left[\log D_\theta\left(G_\phi(z)\right)\right]$$

ただし、この場合は密度比推定に基づく解釈は成り立たない

GANの派生

データ分布との距離の指標をJS以外に変えると、様々なGANの派生系が作れる

例：Wasserstein GAN

$$\mathscr{L}\left(\theta, \phi\right) = \mathbb{E}_{p(x)}\left[D_\theta(x)\right] - \mathbb{E}_{p(z)}\left[D_\theta\left(G_\phi(z)\right)\right]$$

D_θ は1-リプシッツな関数（$\mathbb{R}^{d_x} \to \mathbb{R}$）

このとき、生成器の学習は1-Wasserstein distanceの最小化となる

Energy-based Model

エネルギー関数 $E_\theta(x)$ で確率モデルを表現する

$$p_\theta(x) = \frac{\exp\left(-E_\theta(x)\right)}{Z(\theta)}$$

$$Z(\theta) = \int \exp\left(-E_\theta(x)\right) \mathrm{d}x$$

$E_\theta(x)$ は負の対数尤度 $-\log p_\theta(x)$ に比例

EBMの学習

Contrastive Divergence

EBMの対数尤度の勾配は

$$
\begin{aligned}
\nabla_\theta \log p_\theta(x) &= -\nabla_\theta E_\theta(x) + \nabla_\theta \log Z(\theta) \\
&= -\nabla_\theta E_\theta(x) + \mathbb{E}_{x' \sim p_\theta(x)}\left[\nabla_\theta E_\theta(x')\right]
\end{aligned}
$$

訓練データのエネルギーを下げて、モデルからのサンプルのエネルギーを上げる

EBMからのサンプリング
Langevin dynamics

勾配降下法にノイズがのった形

勾配ベースのMCMC

$$x \leftarrow x - \eta \nabla_x E_\theta[x] + \epsilon$$
$$\epsilon \sim \mathrm{Normal}\left(0, 2\eta I\right)$$

この更新式で繰り返しサンプリングすると、サンプルの分布は $p_\theta(x)$ に収束する

EBMの学習

Contrastive Divergence

まとめると、EBMの更新式は

$$\mathscr{L}(\theta, x') = -\sum_{i=1}^{N} E_\theta\left(x_i\right) + E_\theta\left(x_i'\right)$$

$$\theta \leftarrow \theta + \eta_\theta \nabla_\theta \mathscr{L}(\theta, x')$$

$$x_i' \leftarrow x_i' - \eta_{x'} \nabla_{x'} \mathscr{L}(\theta, x') + \epsilon$$

$$\epsilon \sim \mathrm{Normal}\left(0{,}2\eta I\right)$$

EBMの学習

Contrastive Divergence

まとめると、EBMの更新式は

$$\mathcal{L}\left(\theta, x'\right) = -\sum_{i=1}^{N} E_\theta\left(x_i\right) + E_\theta\left(x'_i\right)$$

$$\theta \leftarrow \theta + \eta_\theta \nabla_\theta \mathcal{L}\left(\theta, x'\right)$$

$$x'_i \leftarrow x'_i - \eta_{x'} \nabla_{x'} \mathcal{L}\left(\theta, x'\right) + \epsilon$$

$$\epsilon \sim \mathrm{Normal}\left(0, 2\eta I\right)$$

よく見るとGANっぽい

GANの更新式

$$\mathscr{L}\left(\theta, \phi\right) = \sum_{i=1}^{N} \log D_\theta(x_i) + \log\left(1 - D_\theta\left(G_\phi\left(z_i\right)\right)\right)$$

$$\theta \leftarrow \theta + \eta_\theta \nabla_\theta \mathscr{L}\left(\theta, \phi\right)$$

$$\phi \leftarrow \phi - \eta_\phi \nabla_\phi \mathscr{L}\left(\theta, \phi\right)$$

$$z_i \sim \mathrm{Normal}\left(0, I\right)$$

GANの更新式 with -logDトリック

$$\mathscr{L}\left(\theta, \phi\right) = \sum_{i=1}^{N} \log D_\theta(x_i) - \log D_\theta\left(G_\phi\left(z_i\right)\right)$$

$$\theta \leftarrow \theta + \eta_\theta \nabla_\theta \mathscr{L}\left(\theta, \phi\right)$$

$$\phi \leftarrow \phi - \eta_\phi \nabla_\phi \mathscr{L}\left(\theta, \phi\right)$$

$$z_i \sim \mathrm{Normal}\,(0, I)$$

GANの更新式 with -logDトリック

$E_\theta(x) = -\log D_\theta(x)$とおくと

$$\mathcal{L}\left(\theta, \phi\right) = -\sum_{i=1}^{N} E_\theta\left(x_i\right) + E_\theta\left(G_\phi\left(z_i\right)\right)$$

$$\theta \leftarrow \theta + \eta_\theta \nabla_\theta \mathcal{L}\left(\theta, \phi\right)$$

$$\phi \leftarrow \phi - \eta_\phi \nabla_\phi \mathcal{L}\left(\theta, \phi\right)$$

$$z_i \sim \mathrm{Normal}\,(0, I)$$

GANとEBMの類似性

GAN with - logD trick

$$\mathscr{L}\left(\theta, \phi\right) = -\sum_{i=1}^{N} E_\theta\left(x_i\right) + E_\theta\left(G_\phi\left(z_i\right)\right)$$

$$\theta \leftarrow \theta + \eta_\theta \nabla_\theta \mathscr{L}\left(\theta, \phi\right)$$

$$\phi \leftarrow \phi - \eta_\phi \nabla_\phi \mathscr{L}\left(\theta, \phi\right)$$

$$z_i \sim \mathrm{Normal}\left(0, I\right)$$

EBM

$$\mathscr{L}\left(\theta, x'\right) = -\sum_{i=1}^{N} E_\theta\left(x_i\right) + E_\theta\left(x_i'\right)$$

$$\theta \leftarrow \theta + \eta_\theta \nabla_\theta \mathscr{L}\left(\theta, x'\right)$$

$$x_i' \leftarrow x_i' - \eta_{x'} \nabla_{x'} \mathscr{L}\left(\theta, x'\right) + \epsilon$$

$$\epsilon \sim \mathrm{Normal}\left(0, 2\eta I\right)$$

めっちゃ似てるけどちょっと違う

GANとEBMの類似性

GAN with - logD trick

$$\mathscr{L}\left(\theta, \phi\right) = -\sum_{i=1}^{N} E_\theta\left(x_i\right) + E_\theta\left(\underline{G_\phi\left(z_i\right)}\right)$$

$$\theta \leftarrow \theta + \eta_\theta \nabla_\theta \mathscr{L}\left(\theta, \phi\right)$$

$$\underline{\phi \leftarrow \phi - \eta_\phi \nabla_\phi \mathscr{L}\left(\theta, \phi\right)}$$

$$z_i \sim \mathrm{Normal}\left(0, I\right)$$

サンプルを直接更新する代わりに

ノイズからサンプルを生成する関数G_ϕを

更新する

EBM

$$\mathscr{L}\left(\theta, x'\right) = -\sum_{i=1}^{N} E_\theta\left(x_i\right) + E_\theta\left(x'_i\right)$$

$$\theta \leftarrow \theta + \eta_\theta \nabla_\theta \mathscr{L}\left(\theta, x'\right)$$

$$x'_i \leftarrow x'_i - \eta_{x'} \nabla_{x'} \mathscr{L}\left(\theta, x'\right) + \underline{\epsilon}$$

$$\epsilon \sim \mathrm{Normal}\left(0, 2\eta I\right)$$

更新にノイズがのる

めっちゃ似てるけどちょっと違う

論文紹介

EBMの学習をGANみたいに生成器を使ってできないか？

➡ 論文1, 2

GANの識別器をエネルギー関数とみなすと、生成時に識別器を使えるのでは？

➡ 論文3

Deep Directed Generative Models with Energy-Based Probability Estimation

https://arxiv.org/abs/1606.03439

Taesup Kim, Yoshua Bengio (Université de Montréal)

EBMの学習

Contrastive Divergence

EBMの対数尤度の勾配

$$\nabla_\theta \log p_\theta(x) = -\nabla_\theta E_\theta(x) + \mathbb{E}_{x' \sim p_\theta(x)}\left[\nabla_\theta E_\theta(x')\right]$$
$$\approx -\nabla_\theta E_\theta(x) + \mathbb{E}_{z \sim p(z)}\left[\nabla_\theta E_\theta\left(G_\phi(z)\right)\right]$$

$p_\theta(x)$ からのサンプリングを $G_\phi(z)$ からのサンプリングで置き換える

生成器の学習

$p_\phi(x) = \mathbb{E}_{p(z)}\left[\delta\left(G_\phi(z)\right)\right]$ とすると、$p_\theta(x) = p_\phi(x)$ となれば良いので

この2つの分布のKL divergenceを最小化することで学習する

$$\mathrm{KL}\left(p_\phi \| \, p_\theta\right) = \underbrace{\mathbb{E}_{p_\phi}\left[-\log p_\theta(x)\right]}_{\text{サンプルのエネルギーを下げる}} - \underbrace{H\left(p_\phi\right)}_{\text{サンプルのエントロピーを上げる}}$$

生成器の学習
なぜエントロピー項が必要か

$$\underbrace{\mathrm{KL}\left(p_\phi \| \, p_\theta\right) = \mathbb{E}_{p_\phi}\left[-\log p_\theta(x)\right]}_{\text{サンプルのエネルギーを下げる}} - \underbrace{H\left(p_\phi\right)}_{\text{サンプルのエントロピーを上げる}}$$

もしエントロピー項がないと、生成器はエネルギーが最小（＝密度が最大）のサンプルのみを生成するように学習してしまう

- GANのmode collapseと似たような現象

これを防ぐためにエントロピー項が必要

生成器の学習

第1項の勾配は、以下のように簡単に計算可能

$$\nabla_\phi \mathbb{E}_{p_\phi}\left[-\log p_\theta(x)\right] = \mathbb{E}_{z\sim p(z)}\left[\nabla_\phi E_\theta\left(G_\phi(z)\right)\right]$$

生成器の学習

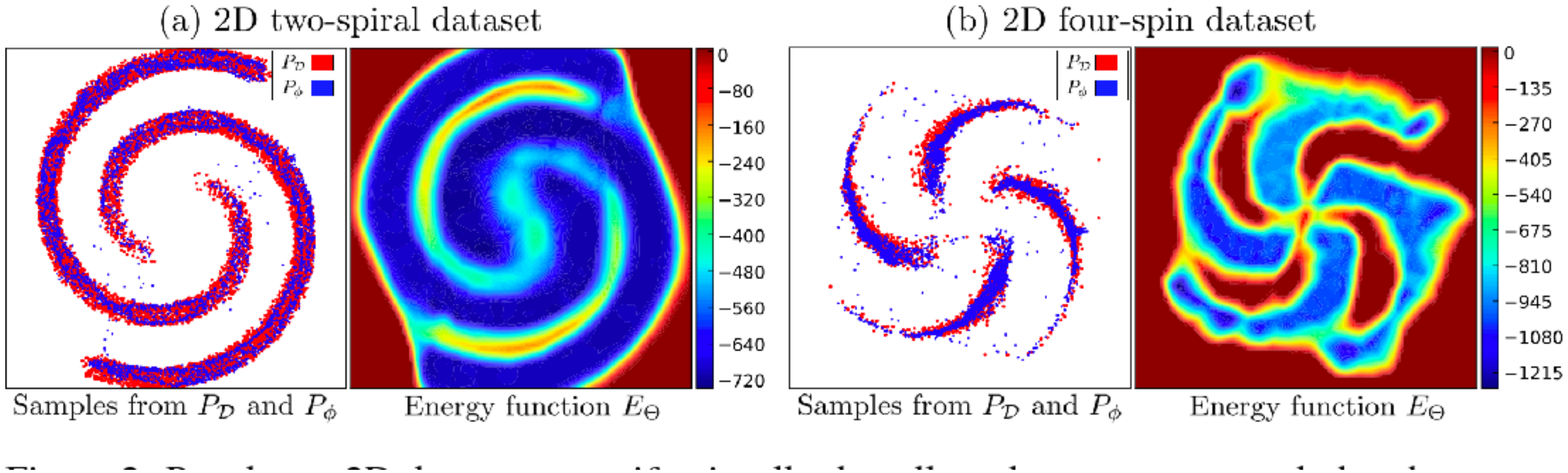
第2項のエントロピーは解析的に求まらない

論文では、バッチ正規化のスケールパラメータを正規分布の分散とみなして
そのエントロピーを計算することで代用している

$$H\left(p_\phi\right) \approx \sum_{a_i} H\left(\mathcal{N}\left(\mu_{a_i}, \sigma_{a_i}\right)\right) = \sum_{a_i} \frac{1}{2} \log\left(2e\pi\sigma_{a_i}^2\right)$$

GANに対する利点

識別器の代わりにエネルギー関数を学習するので、密度比推定などに使える



生成サンプル

Maximum Entropy Generators for Energy-Based Models

https://arxiv.org/abs/1901.08508

Rithesh Kumar, Sherjil Ozair, Anirudh Goyal, Aaron Courville, Yoshua Bengio (Université de Montréal)

エントロピーの計算

$$\mathrm{KL}\left(p_\phi \| \, p_\theta\right) = \mathbb{E}_{p_\phi}\left[-\log p_\theta(x)\right] - \mathrm{H}\left(p_\phi\right)$$

論文1ではエントロピー$H\left(p_\phi\right)$の計算をバッチ正規化のスケールパラメータで行っていたが、ヒューリスティックで理論的な妥当性もない

エントロピーの計算

潜在変数 z と生成器の出力 $x = G_\phi(z)$ の相互情報量を考えると

$$I(x, z) = H(x) - H\left(x \mid z\right) = \mathbb{E}_{p(z)}\left[H\left(G_\phi(z)\right) - H\left(G_\phi(z) \mid z\right)\right]$$

エントロピーの計算

G_ϕ が決定論的な関数のとき、$H\left(G_\phi(z) \mid z\right) = 0$ なので

$$H\left(p_\phi\right) = \mathbb{E}_{p(z)}\left[H\left(G_\phi(z)\right)\right] = I(x, z)$$

つまり、エントロピーの代わりに、相互情報量を最大化すれば良い

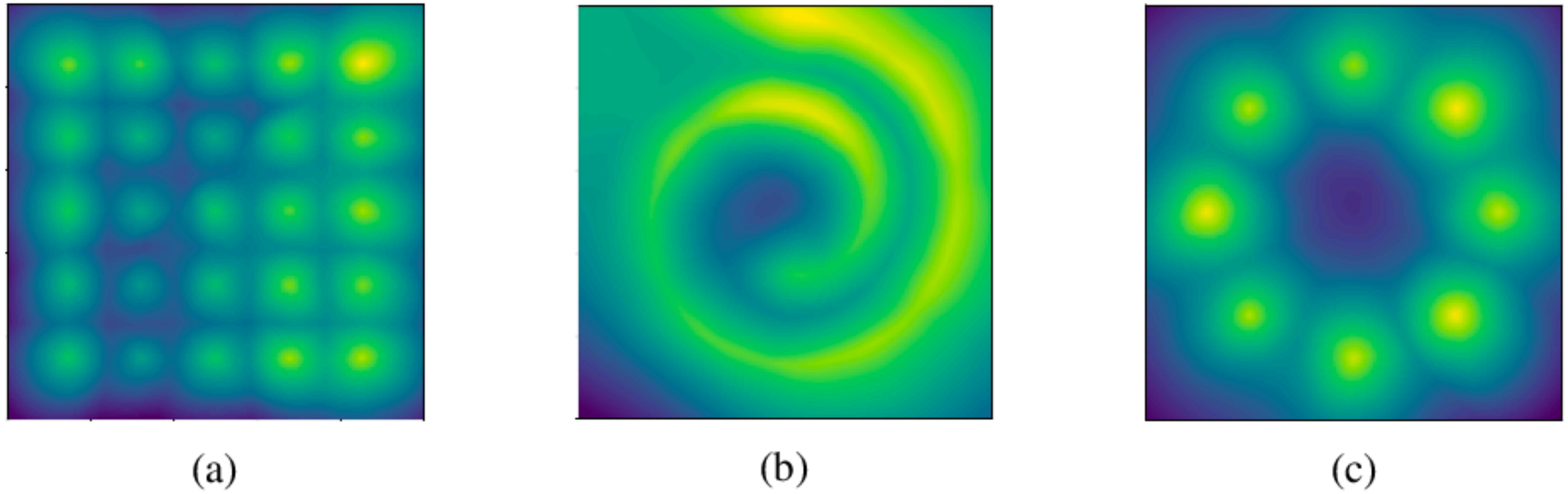
相互情報量の推定

相互情報量の推定方法は、近年いろいろ提案されているが、
ここでは、Deep InfoMaxで提案されたJS divergenceに基づく推定法を用いる

$$I_{\mathrm{JSD}}(x, z) = \sup_{T \in \mathcal{T}} \mathbb{E}_{p(x, z)} \left[-\mathrm{sp}\left(-T(x, z) \right) \right] - \mathbb{E}_{p(x)p(z)} \left[\mathrm{sp}\left(T(x, z) \right) \right]$$

T は $p(x, z)$ からのサンプルと $p(x)\,p(z)$ からのサンプルを見分ける識別器で
同時に学習する

密度推定



Probability density visualizations for three popular toy dataset (a) 25 gaussians, (b)

複雑な分布もうまく近似できている

Mode Collapse

1000 (or 10000) 個のモードをもつデータ（StackedMNIST）で学習したときに、モードをいくつ捉えられているかを比較する実験

MEG (提案法) はすべてのモードを捉えており、mode collapseが起こらない

are borrowed from Belghazi et al. (2018)

(Max 10^3)	Modes	KL
Unrolled GAN	48.7	4.32
VEEGAN	150.0	2.95
WGAN-GP	959.0	0.7276
MEG (ours)	1000.0	0.0313

(Max 10^4)	Modes	KL
WGAN-GP	9538.0	0.9144
MEG (ours)	10000.0	0.0480

画像生成

CIFAR-10

EBMからMCMCでサンプルした場合はWGAN-GPよりもIS, FIDが良い

Method	**Inception score**	**FID**
Real data	11.24±.12	7.8
WGAN-GP	6.81 ± .08	30.95
MEG (Generator)	6.49 ± .05	35.02
MEG (MCMC)	**7.31 ± .06**	33.18

Your GAN is Secretly an Energy-based Model and You Should Use Discriminator Driven Latent Sampling

https://arxiv.org/abs/2003.06060

Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, Yoshua Bengio (Université de Montréal, Google Brain)

GANの一般的な解釈
識別器＝密度比推定器

識別器はデータ分布 $p(x)$ と生成サンプルの分布 $p_\phi(x) = \mathbb{E}_{p(z)}\left[p\left(G_\phi(z)\right)\right]$ の密度比推定器としての役割を果たす

i.e., 識別器が最適なとき

$$D_\theta^*(x) = \frac{p(x)}{p(x) + p_\phi(x)}$$

GANの一般的な解釈

識別器＝密度比推定器

$\sigma(\cdot)$ をシグモイド関数、$d_\theta(x) = \sigma^{-1}(D(x))$ とすると

$$D_\theta(x) = \frac{p(x)}{p(x) + p_\phi(x)} \quad \Rightarrow \quad p(x) \propto p_\phi(x) \exp\left(d_\theta(x)\right)$$

データ分布 $p(x)$ は生成器の分布 $p_\phi(x)$ と $\exp\left(d_\theta(x)\right)$ の積に比例する

➡ 学習後のGANでこの分布からサンプルすれば、生成の質が上がるのでは？

潜在空間でのMCMC

Discriminator Driven Latent Sampling (DDLS)

$p_\phi(x)\exp\left(d_\theta(x)\right)$ からサンプリングしたいが、データ空間でMCMCをするのは効率が悪く難しい

代わりに生成器 $G_\phi(z)$ の潜在空間上でMCMC (Langevin dynamics)を行う

$$E(z) = -\log p(z) - d_\theta\left(G_\phi(z)\right)$$

$$z \leftarrow z - \eta \nabla_z E(z) + \epsilon$$

$$\epsilon \sim \mathrm{Normal}\left(0, 2\eta I\right)$$

実験

学習済みのGANにDDLSを使うだけで、ISやFIDがかなり改善する

Model	Inception	FID
PixelCNN (van den Oord et al., 2016)	4.60	65.93
PixelIQN (Ostrovski et al., 2018)	5.29	49.46
EBM (Du & Mordatch, 2019)	6.02	40.58
WGAN-GP (Gulrajani et al., 2017)	$7.86 \pm .07$	36.4
MoLM (Ravuri et al., 2018)	$7.90 \pm .10$	18.9
SNGAN (Miyato et al., 2018)	$8.22 \pm .05$	21.7
ProgressiveGAN (Karras et al., 2018)	$8.80 \pm .05$	-
NCSN (Song & Ermon, 2019)	$8.87 \pm .12$	25.32
DCGAN w/o DRS or MH-GAN	2.8789	-
DCGAN w/ DRS(cal) (Azadi et al., 2018)	3.073	-
DCGAN w/ MH-GAN(cal) (Turner et al., 2019)	3.379	-
ResNet-SAGAN w/o DOT	$7.85 \pm .11$	21.53
ResNet-SAGAN w/ DOT	$8.50 \pm .12$	19.71
SNGAN w/o DDLS	$8.22 \pm .05$	21.7
Ours: SNGAN w/ DDLS	$9.05 \pm .11$	15.76
Ours: SNGAN w/ DDLS(cal)	$\mathbf{9.09 \pm 0.10}$	**15.42**

まとめ

GANとEBMは深い関係にある

両者の知見を生かすことで、両者のいいとこ取りをするアプローチができる

- EBMのサンプリングに生成器を使う
- GANのサンプリングにMCMCを使う

今後も似たようなアプローチの研究が色々と出てくる予感