Graphical Generative Adversarial Networks

Chongxuan Li[1],Max Welling[2],Jun Zhu[1],Bo Zhang[1]
1: Tsinghua University, 2: University of Amsterdam
Neural Information Processing Systems (NeurIPS 2018),

Presenter: Arshdeep Sekhon
https://qdata.github.io/deep2Read

Overview

Motivation: Generative Models

Given true $p(x)$, two ways to model data distribution:

- **Prescribed Probability Models**: Specify log likelihood $q_\theta(x)$, maximize to find θ, indexing a family of possible distributions

Motivation: Generative Models

Given true $p(x)$, two ways to model data distribution:

- **Prescribed Probability Models**: Specify log likelihood $q_\theta(x)$, maximize to find θ, indexing a family of possible distributions
- **Implicit Probabilistic Models:**Map z from an easy to sample data distribution, $G : Z \to X$ paramterized by θ

Motivation: Generative Models

Deep Implicit vs Probabilistic Graphical Models

- Deep Implicit Models: do not model structure in data
- Probabilistic Graphical Models: can model prior knowledge about data but can't deal with images

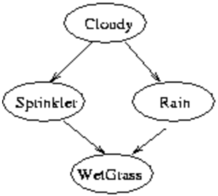
Deep Implicit Models

- only a simulation of the generative process without explicit likelihood evaluation
- density $q_\theta(x)$ can be highly intractable:

$$q_\theta(x) = \frac{\partial}{\partial x_1} \cdots \frac{\partial}{\partial x_d} \int_{\{z : \mathcal{G}_\theta(z) \leq x\}} q(z)\, dz$$

Probabilistic Graphical Models

For example, Bayesian Networks



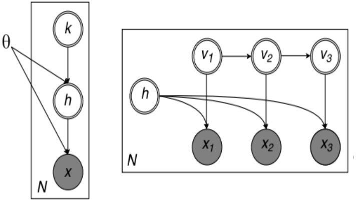
Bayes Net

Bayesian network joint distribution

a node is independent of its ancestors given its parents.

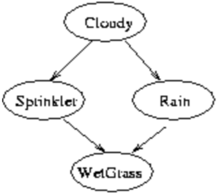
This Paper: Graphical GAN

- combine both Deep Implicit vs Probabilistic Graphical Models
- Representation of variables: Bayesian Network
- Probabilistic Modeling: Deep Implicit likelihood function
- Structure:



Incorporating Structure: Probabilistic Inference

Given x,what z is likely to have produced it?



Bayes Net

Inference: In the water sprinkler network, and suppose we observe the fact that the grass is wet. There are two possible causes for this: either it is raining, or the sprinkler is on. Which is more likely?

- Can be done in probabilistic graphical models

Graphical GANs: Model Definition: $P_G(X, Z)$

- Structured Data from a Bayes Network G directed acyclic graph
- Can write $P_G(X, Z)$ as:

$$p_G(X, Z) = \prod_{i=1}^{|Z|} p(z_i | pa_G(z_i)) \prod_{j=1}^{|Z|} p(x_j | pa_G(x_j)) \tag{1}$$

- easy to sample from using ancestral sampling
- Parametrize the dependency functions as DNNs

Graphical GANs

Two issues:

- deep implicit likelihood functions: makes the inference of the latent variables intractable

Graphical GANs

Two issues:

- deep implicit likelihood functions: makes the inference of the latent variables intractable
- complex structures: which requires the inference and learning algorithm to exploit the structural information explicitly

Learning: Adversarial Learned Inference

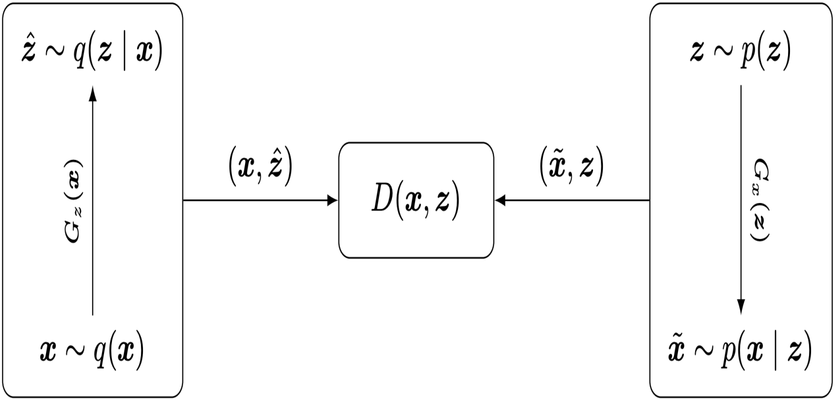
- GANs can't do inference
- BiGANs introduced to do inference

$$min_{\theta,\phi} D(q(X, Z)||p(X, Z)) \tag{2}$$

where D is in the f-divergence family

- cannot optimize directly: likelihood ratio is unknown given implicit p(X, Z)

Learning: Adversarial Learned Inference(BiGAN)



Learning: Extending to Structured Data

- Extend BiGAN ALI to Graphical GANs
- Given $P_G(X, Z)$ and $Q_G(X, Z)$
- Discriminator that takes in both (X,Z)

Learning with Structured Data: Expectation Propagation

- factorization of $p(X, Z)$ in terms of a set of factors F_G
- $p(X, Z) \propto \prod_{A \in F_G} p(A)$
- Similarly for $Q(X, Z)$
- $q(X, Z) \propto \prod_{A \in F_G} q(A)$

- EP iteratively minimizes a local divergence in terms of each factor individually.
- for factor A:

$$D(q(A)\overline{q(A)})||D(p(A)\overline{p(A)}) \tag{3}$$

- $\overline{p(A)}$ denotes the marginal distribution over the complementary $(\overline{A})$ of A.
- Assume $\overline{p(A)} \approx \overline{q(A)}$

Learning with Structured Data: Expectation Propagation

$$
\begin{aligned}
&\mathcal{D}_{JS}(q(X,Z)||p(X,Z)) \\
\approx&\mathcal{D}_{JS}(q(A)\overline{q(A)}||p(A)\overline{p(A)}) \\
\approx&\mathcal{D}_{JS}(q(A)\overline{q(A)}||p(A)\overline{q(A)}) \\
=&\int q(A)\overline{q(A)}\log\frac{q(A)\overline{q(A)}}{\frac{p(A)\overline{q(A)}+q(A)\overline{q(A)}}{2}}dXdZ+\int p(A)\overline{q(A)}\log\frac{p(A)\overline{q(A)}}{\frac{p(A)\overline{q(A)}+q(A)\overline{q(A)}}{2}}dXdZ \\
=&\int q(A)\overline{q(A)}\log\frac{q(A)}{m(A)}dXdZ+\int p(A)\overline{q(A)}\log\frac{p(A)}{m(A)}dXdZ \\
\approx&\int q(A)\overline{q(A)}\log\frac{q(A)}{m(A)}dXdZ+\int p(A)\overline{p(A)}\log\frac{p(A)}{m(A)}dXdZ \\
\approx&\mathbb{E}_q\log\frac{q(A)}{m(A)}+\mathbb{E}_p\log\frac{p(A)}{m(A)},
\end{aligned}
$$

Learning with Structured Data: Expectation Propagation

$$\frac{1}{|F_{\mathcal{G}}|}\sum_{A\in F_{\mathcal{G}}}\left[\mathbb{E}_q[\log\frac{q(A)}{m(A)}]+\mathbb{E}_p[\log\frac{p(A)}{m(A)}]\right]=\frac{1}{|F_{\mathcal{G}}|}\left[\mathbb{E}_q[\sum_{A\in F_{\mathcal{G}}}\log\frac{q(A)}{m(A)}]+\mathbb{E}_p[\sum_{A\in F_{\mathcal{G}}}\log\frac{p(A)}{m(A)}]\right].$$

average the divergences over all local factors

$$\max_{\psi}\frac{1}{|F_{\mathcal{G}}|}\mathbb{E}_q[\sum_{A\in F_{\mathcal{G}}}\log(D_A(A))]+\frac{1}{|F_{\mathcal{G}}|}\mathbb{E}_p[\sum_{A\in F_{\mathcal{G}}}\log(1-D_A(A))],$$

D_A is the discriminator for the factor A and ψ denotes the parameters in all discriminators

Inference

Given x,what z is likely to have produced it?

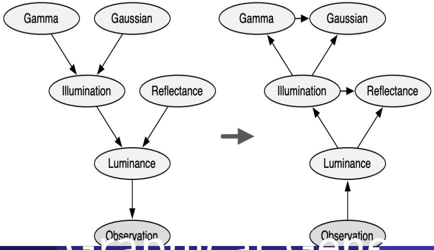
- Consider structure of graphical model while doing $Q(X, Z)$
- Two ways:
  - Mean Field Posteriors
  - Inverse Factorization

Mean Field Propagation

- $q_H(X, Z) = q(X)q_H(Z|X)$
- all of the dependency structures among the latent variables are ignored
- $$q_H(Z|X) = \prod_{i=1}^{|Z|} q(z_i|X) \tag{4}$$
- where the associated graph H has fully factorized structures.

Inverse Factorizations

- sample the latent variables given the observations efficiently by inverting G step by step.
- $$q_H(Z|X) = \prod_{i=1}^{|Z|} q(z_i|\partial_G(z_i) \cap z_{>i}) \quad (5)$$
- Given the structure of the approximate posterior: parameterize the dependency functions as neural networks of similar sizes to those in the generative models



Training Algorithm

Algorithm 1 Local algorithm for Graphical-GAN

repeat

- Get a minibatch of samples from $p(X, Z)$
- Get a minibatch of samples from $q(X, Z)$
- Approximate the divergence $\mathcal{D}(q(X, Z)||p(X, Z))$ using Eqn. (12) and the current value of ψ
- Update ψ to maximize the divergence
- Get a minibatch of samples from $p(X, Z)$
- Get a minibatch of samples from $q(X, Z)$
- Approximate the divergence $\mathcal{D}(q(X, Z)||p(X, Z))$ using Eqn. (12) and the current value of ψ
- Update θ and ϕ to minimize the divergence

until Convergence or reaching certain threshold

Instance 1: GM-GAN

- assume that the data consists of K mixtures and hence uses a mixture of Gaussian prior
- $k \sim Cat(\pi), h|k \sim N(\mu_k, \Sigma_k), x|h = G(h)$
- π and $Sigma_k$s are fixed as the uniform prior and identity matrices
- Inverse factorization

$$h|x = E(x); q(k|h) = \frac{\pi_k N(h|\mu_k, \Sigma_k)}{\Sigma'_k \pi'_k N(h|\mu'_k, \Sigma'_k)} \tag{6}$$

- In the global baseline, a single network is used to discriminate the (x, h, k) tuples.
- local algorithm: two separate networks are introduced to discriminate the (x, h) and (h, k) pairs, respectively.

Instance 2: StateSpace-GAN

- two types of latent variables: One is invariant across time h and the other varies across time v_t for time stamp $t = 1, \ldots, T$
- use the mean-field recognition model as the approximate posterior:

$$v_1 \sim \mathcal{N}(0, I), h \sim \mathcal{N}(0, I), \qquad \epsilon_t \sim \mathcal{N}(0, I), \forall t = 1, 2, ..., T-1,$$
$$v_{t+1}|v_t = O(v_t, \epsilon_t), \forall t = 1, 2, ..., T-1, \qquad x_t|h, v_t = G(h, v_t), \forall t = 1, 2, ..., T,$$

$$h|x_1, x_2..., x_T = E_1(x_1, x_2..., x_T), \qquad v_t|x_1, x_2..., x_T = E_2(x_t), \forall t = 1, 2, ..., T,$$

Instance 1: GMGAN Learns Discrete Structures

Assumption : that there exist discrete structures, e.g. classes and attributes, in the data but the ground truth is unknown

(a) GAN-G (b) GMGAN-G ($K = 10$) (c) GMGAN-L ($K = 10$) (d) GMVAE ($K = 10$)

GMGAN Learns Discrete Structures

(a) $(K = 50)$ (b) $(K = 30)$ (c) $(K = 100)$

GMGAN Learns Discrete Structures

Algorithm	ACC on MNIST	IS on CIFAR10	MSE on MNIST
GMVAE	92.77 (±1.60) [7]	-	-
CatGAN	90.30 [37]	-	-
GAN-G	-	5.34 (±0.05) [45]	-
GMM (our implementation)	68.33(±0.21)	-	-
GAN-G+GMM (our implementation)	70.27(±0.50)	5.26 (±0.05)	0.071 (±0.001)
GMGAN-G (our implementation)	91.62 (±1.91)	5.41 (±0.08)	0.056 (±0.001)
GMGAN-L (ours)	**93.03** (±1.65)	**5.94** (±0.06)	**0.044** (±0.001)

Instance 2: SSGAN Learns Temporal Structures

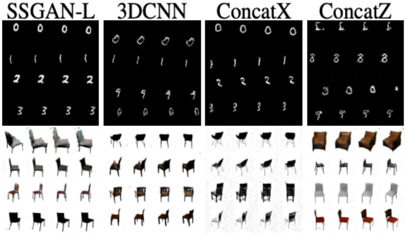


Figure 5: Samples on the Moving MNIST and 3D chairs datasets when $T = 4$. Each row in a subfigure represents a video sample.

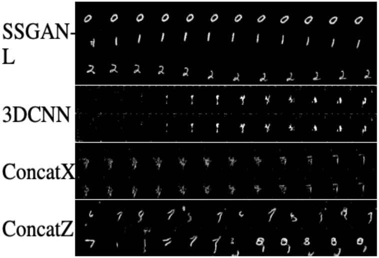


Figure 6: Samples (first 12 frames) on the Moving MNIST dataset when $T = 16$.

Figure 7: Motion analogy results. Each odd row shows an input and the next even row shows the sample.

Figure 8: 16 of 200 frames generated by SSGAN-L. The frame indices are 47-50, 97-100, 147-150 and 197-200 from left to right in each row.

Summary

- utilize the underlying structural information of the data in an implicit likelihood setting
- learning interpretable representations and generating structured samples
- Not generalized
- More Complicated Structures?