Analysing Failure Modes in Unsupervised Image-to-Image Translation

Doctoral Qualifying Oral Exam
Boston University 2021
Ben Usman
Presentation Plan

1. Problem and motivation
2. Existing solutions and possible failure modes
3. Tools for analysing these failure modes in prior work
   a. “Generalization and Equilibrium in Generative Adversarial Nets”
      by Arora et al., PMLR 2017.
   b. “Training Generative Adversarial Networks with Limited Data”
      by Karras et al., NeurIPS 2020.
   c. “Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”
      by Galanti et al., JMLR 2021. // 2017-2021
Task

Source Samples (Cats)

Target Samples (Dogs)

✓ is a dog
✓ same coat color
✓ same pose
...

F
Ground truth 1-to-1 cross-domain mapping.
Task

Source Samples

Goal: reconstruct F from unpaired samples

Target Samples
Task

1D manifold

2D manifold
How to find a good $F$?

Source Samples

Translated Source Samples

Target Samples

$A = \{a_i\}$

$F(A) = \{F(a_i) : a_i \in A\}$

$B = \{b_j\}$

$\min_{F \in \mathcal{F}} \ d(F(A), B) + R(F)$
How to find a good $F$? - what we expect

$F_{t=0}$

Translated Source Samples

HIGH statistical distance $d(F(A), B)$

Target Samples

$F_{t=T}$

LOW statistical distance $d(F(A), B)$

$\min_{F \in F} d(F(A), B) + R(F')$
Why care about this problem?

This is an unsupervised generative problem that has GT outputs!

As a result, we are learning a neural data model, but can reason about its correctness and the prediction error vs GT outputs (e.g. L2).

In contrast,
- in GANs - there are no expected outputs
- in classification/regression - often no need to model data.
Presentation Plan

1. Problem and motivation
2. Existing solutions and possible failure modes
3. Tools for analysing these failure modes in prior work
   a. “Generalization and Equilibrium in Generative Adversarial Nets”
      by Arora et al., PMLR 2017.
   b. “Training Generative Adversarial Networks with Limited Data”
      by Karras et al., NeurIPS 2020.
   c. “Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”
      by Galanti et al., JMLR 2021. // 2017-2021
What could go wrong?

I: The statistical distance is too weak

Source Samples

Target Samples

\[ A = \{a_i\} \]

\[ F(A) = \{F(a_i) : a_i \in A\} \]

\[ B = \{b_i\} \]

\[ \min_{F \in \mathcal{F}} d(F(A), B) + R(F) \]
What could go wrong?
I: The statistical distance is too strong

- AS HIGH AS JUST NOISE
- LOW statistical distance

\[ d(F(A), B) \]
What could go wrong?

II: The stat distance is **too sharp** (hard to optimize)

\[ d(F(A), B) \]

**EQUALLY HIGH**

statistical distance
What could go wrong?

III: The final mapping is nonsensical

\[
\min_{F \in \mathcal{F}} d(F(A), B) + R(F)
\]
Selected prior work


Why these papers?

Introduce important theoretical tools to understand related problems:

- **Chernoff bound and \( \varepsilon \)-cover method** to estimate sample complexity of adversarial statistical distances
- **\( \varepsilon \)-approximate Nash equilibrium** to analyse the existence of the solution to the adversarial alignment problem
- **Markov operators and group structure of augmentations** to estimate statistical distances between distributions under data augmentations
- **unsupervised bias-variance tradeoff and Rademacher complexity** to relate the prediction error with the alignment error and the complexity of the function class
Other background papers

I also use there papers / books to provide context / refer for proofs

- “Simple Strategies for Large Zero-Sum Games with Applications to Complexity Theory” by Lipton & Young, STOC’94
- “Towards Principled Methods for Training Generative Adversarial Networks”, Arjovsky & Bottou, ICLR’17
- “Stabilizing Training of Generative Adversarial Networks through Regularization”, Roth et al, NeurIPS’17
- “Which Training Methods for GANs do actually Converge?”, Mescheder et al., ICML’18
- “Kernel of CycleGAN as a Principle homogeneous space”, Moriakov et al., ICLR’20
- “Guiding the One-to-One Mapping in CycleGAN via Optimal Transport’, Lu et al., AAAI’19
Other background papers

I have “backup slides” covering these papers as well in the end:

- "Table for estimating the goodness of fit of empirical distributions", Smirnov, Annals of Mathematical Statistics ’48 - introduces KS-test
- “A Kernel Two-Sample Test”, Gretton et al, JMLR’12 - introduces MMD test
- “Wasserstein GAN” Arjovsky et al, ICML’17 - introduces WGAN objective
- “Are GANs Created Equal? A Large-Scale Study”, Lucic et al., NeurIPS’18; “Pros and Cons of GAN Evaluation Measures”, Ali Borji, arxiv’18; “Improved Precision and Recall Metric for Assessing Generative Models”, Kynkänniemi et al., NeurIPS’19 - introduces FID, KID, IS, GAN-F1 score and compares them
Other background papers

I have “backup slides” covering these papers as well in the end:

- “On the Decreasing Power of Kernel and Distance based Nonparametric Hypothesis Tests in High Dimensions”, Ramdas et al., AAAI’15 - shows that with a “fair alternative” MMD test has exponentially low power in higher dims
- “Revisiting Classifier Two-Sample Tests”, Lopez-Paz et al., ICLR’17 - compares the test power of the GAN-like objective to MMD/KS/other test
- “Reducing Noise in GAN Training with Variance Reduced Extragradient”, Chavdarova et al., NeuIPS’19
How to choose the statistical distance?

**Def 1:** the statistical distance \(d(A, B)\) “generalizes”

\[
|d(\mathcal{D}_{\text{real}}, \mathcal{D}_G) - d(\hat{\mathcal{D}}_{\text{real}}, \hat{\mathcal{D}}_G)| \leq \varepsilon \quad \text{(with exponentially high probability over the choice of } m \text{ samples as the number of samples increases)}
\]

**Lem 1:** JSD and Wasserstein distances “do not generalize”!

\[
\mathcal{N}(0, \frac{1}{d} I) = \mu \implies d_{JS}(\mu, \hat{\mu}) = \log 2, \quad d_W(\mu, \hat{\mu}) \geq 1.1
\]

\[
\text{JS}(p; q) = \frac{1}{2} \int \left(p \log \frac{2p}{p+q} + q \log \frac{2q}{p+q}\right) d\mu
\]

(for \(q = 0\) almost everywhere)

\[
\Pr[\forall i \in [m]|y - x_i| \geq 1.2] \geq 1 - m \exp(-\Omega(d)) \geq 1 - o(1)
\]

\[
d_W(\mu, \hat{\mu}) \geq 1.2 \Pr[\forall i \in [m]|y - x_i| \geq 1.2] \geq 1.1
\]

(prob of all pairs of points in A and B being at least 1.2 away from each other does not decay fast enough with } m)
How to choose the statistical distance?

**Def 2:** $F$-divergence wrt $\phi$

$$d_{\mathcal{F},\phi}(\mu, \nu) = \sup_{D \in \mathcal{F}} \mathbb{E}_{x \sim \mu} [\phi(D(x))] + \mathbb{E}_{x \sim \nu} [\phi(1 - D(x))] - 2\phi(1/2)$$

**Lem 2:**

$$m \geq \frac{cp\Delta^2 \log(LL_{\phi}p/e)}{\varepsilon^2}, \text{ we have with probability at least } 1 - \exp(-p)$$

$$|d_{\mathcal{F},\phi}(\hat{\mu}, \hat{\nu}) - d_{\mathcal{F},\phi}(\mu, \nu)| \leq \varepsilon$$

**Proof:**

a) all $D$ weights can be approx. with err. $< \varepsilon$

using a “covert set” of size $\sim K = \lceil \log(1/e) / e \rceil$

$=>$ worst cast $D$ approx. error $< \varepsilon \cdot L$

b) for any “single fixed” $D$ the est. error $< A$

[“Generalization and Equilibrium in Generative Adversarial Nets” by Arora et al., PMLR 2017]
How to choose the statistical distance?

Def 2: $F$-divergence wrt $\phi$

$$d_{\mathcal{F},\phi}(\mu, \nu) = \sup_{D \in \mathcal{F}} \mathbb{E}_{x \sim \mu} [\phi(D(x))] + \mathbb{E}_{x \sim \nu} [\phi(1 - D(x))] - 2\phi(1/2)$$

Lem 2: $$m \geq \frac{c p \Delta^2 \log(LL_{ph}/\epsilon)}{\epsilon^2}$$, we have with probability at least $1 - \exp(-p)$

$$|d_{\mathcal{F},\phi}(\hat{\mu}, \hat{\nu}) - d_{\mathcal{F},\phi}(\mu, \nu)| \leq \epsilon$$

"Generalization and Equilibrium in Generative Adversarial Nets" by Arora et al., PMLR 2017
How to choose the statistical distance?

**Def 2:** $F$-divergence wrt $\phi$

$$d_{\mathcal{F},\phi}(\mu, \nu) = \sup_{D \in \mathcal{F}} \mathbb{E}[\phi(D(x))] + \mathbb{E}[\phi(1-D(x))] - 2\phi(1/2)$$

**Lem 2:** $m \geq \frac{cp\Delta^2 \log(LL\phi p/\epsilon)}{\epsilon^2}$, we have with probability at least $1 - \exp(-p)$

$$|d_{\mathcal{F},\phi}(\hat{\mu}, \hat{\nu}) - d_{\mathcal{F},\phi}(\mu, \nu)| \leq \epsilon$$

**$\epsilon$-net method on discr weights**

$|X| = O\left(\frac{d \log d}{\epsilon}\right)$

within distance $\epsilon/8LL\phi$

$\log |X| \leq O(p \log(LL\phi p/\epsilon))$

[“Generalization and Equilibrium in Generative Adversarial Nets” by Arora et al., PMLR 2017]
Does the minimum exist?

\[
\min_F \max_D D(X) - D(F(Y))
\]

\[
\min h(F)
\]

\[
\begin{pmatrix}
2 & -2 \\
-1 & 1 \\
\end{pmatrix}
\]

The “pure” equilibrium might not always exist, but a mixed strategy that yields an equilibrium always exist! [Nash ‘50; Glicksberg ‘52]
Def 3: $\epsilon$-approximate equilibrium

$$\forall u \in \mathcal{V}, \quad \mathbb{E}_{u \sim S_u} [F(u, v)] \leq V + \epsilon;$$
$$\forall u \in \mathcal{U}, \quad \mathbb{E}_{v \sim S_v} [F(u, v)] \geq V - \epsilon.$$  

Th: if $p$-parameter ($k$-1)-layer network can generate/discriminate each sample $\Rightarrow \exists k$-layer $G$ and $D$ with $A$ parameters that are in $\epsilon$-eq

Proof:

1) an infinite mixture of $G_i(z) = x_i$, $x_i \sim P(X)$ is mixed Nash eq.
2) K-sized epsilon-net over samples $x_i$ and params of $D$ gives small error $\Rightarrow$ “subsampled” $G'$ and $D'$ are in $\epsilon/2$-eq
3) can approximate “subsampled” $G'(x)$ with a neural $G''(x)$ that “mixes” outputs of $G_i$ with weights produced by a neural $\epsilon/2$-approx. “1-vs-K indicator” $h(z)$, i.e. $G''(x) = \sum_i G_i(z) * h_i(z)$

[“Generalization and Equilibrium in Generative Adversarial Nets” by Arora et al PMLR 2017]
[“Simple Strategies for Large Zero-Sum Games with Applications to Complexity Theory” by Lipton & Young, STOC’94]
Proof:

1) an infinite mixture of $G_i(z) = x_i$ is Nash equilibrium

2) $\varepsilon/4L\ell'L_{\phi}$-net (of size $T$) over params of $G$ and $D$ gives (with high prob) error $<\varepsilon/2 \Rightarrow$ “subsampled” $G'$ and $D'$ are in $\varepsilon$-eq

3) a 2-layer network $h(z)$ can $\delta$-approx. a “multi-way step fn”

4) we build new $G$ that “mixes” outputs of $G'$ with weights produced by $h(z)$, it is $\varepsilon/2$-away from “true mixture of $G_i$’s”

\[
F^*(G, D') \geq \mathbb{E}_{i \in [T], v \in D'} F(u_i, v) - |F^*(G, D') - \mathbb{E}_{i \in [T], v \in D'} F(u_i, v)| \geq V - \varepsilon/2 - 2\Delta \frac{\varepsilon}{4\Delta} \\
F^*(G', D) \leq \mathbb{E}_{i \in [T], u \in G'} F(u, v_i) + |F^*(G', D) - \mathbb{E}_{i \in [T], u \in G'} F(u, v_i)| \leq V + \varepsilon/2 + 2\Delta \frac{\varepsilon}{4\Delta} \\
\] 

second half of the proof is based on

[“Simple Strategies for Large Zero-Sum Games with Applications to Complexity Theory” by Lipton & Young, STOC’94]
Takeaway

1. The “sample GAN loss” reasonably quickly converges to its “true” value.
2. Jensen-Shannon and Wasserstein distances do not.
3. For large networks $\epsilon$-approximate equilibriums exists.

Comments

1. No point in approximating JSD, Wasserstein (and MMD) precisely because their sample estimates are too far from actual values!
2. No point in using them for evaluation either (in higher dimensions)!
3. We are still optimizing for “exact” not “$\epsilon$-approximate” equilibriums.
4. Those equilibriums might also be very hard to get into!

[“Generalization and Equilibrium in Generative Adversarial Nets” by Arora et al., PMLR 2017]
GAN-loss is pretty bad optimization-wise

(with generator fixed after X epochs)

But noise might help:

\[ W(\mathbb{P}_r, \mathbb{P}_g) \leq 2V^{\frac{1}{2}} + 2C\sqrt{JSD(\mathbb{P}_{r+\epsilon} || \mathbb{P}_{g+\epsilon})} \]

[“Towards Principled Methods for Training Generative Adversarial Networks”, Arjovsky & Bottou, ICLR’17]
Instance noise in the discriminator might help. Closed-form regularizer exist.

but requires figuring out a good annealing schedule

[“Stabilizing Training of Generative Adversarial Networks through Regularization”, Roth et al, NeurIPS’17]
[“Instance Noise: A trick for stabilising GAN training”, Ferenc Huszár, inference.vc]
A “toy GAN problem” confirms it.

[“Which Training Methods for GANs do actually Converge?” , Mescheder et al., ICML’18]
Let’s extend to arbitrary augmentations.

Assume augmentation $T(x)$ randomly flips an image by $[0, 90, 180, 270]$ and we apply $T(x)$ “as instance noise” before passing them to $D(x)$ to make images “less separable”.

[“Training Generative Adversarial Networks with Limited Data”, Karras et al., NeurIPS’20]
Here is what you get - “leaking augmentation”.

- $T(x)$ is flip
- $T(x)$ is color shift
How to avoid “leaking augmentation”?

We want $T(x)$ such that $T(P) = T(Q) \iff P = Q$, i.e. we want an invertible operator “$T$: distribution $\overset{\leftrightarrow}{\rightarrow}$ distribution”.

Not same as an invertible augmentation $T(x)$!
Example: $T(P) = P \ast \text{Gaussian}(0, 1)$, i.e. $T(x) = x + \epsilon$, $\epsilon \sim N(0, 1)$.

[“Training Generative Adversarial Networks with Limited Data”, Karras et al., NeurIPS’20]
Markov operator

X = 1D random variable, supp(X) = \{0, 1, 2\}
P(X) = a vector in \(\mathbb{R}^3\) that lies inside \(\Delta_3\)

In this case, the “\(T: \text{distribution} \rightarrow \text{distribution}\)” is just a linear operator “\(T: \Delta_3 \rightarrow \Delta_3\)”.

\[
p = (0.1, 0.1, 0.8)
\]
\[
q = (0.8, 0.1, 0.1)
\]

\[
T = \begin{pmatrix}
0.6 & 0.2 & 0.2 \\
0.2 & 0.6 & 0.2 \\
0.2 & 0.2 & 0.6
\end{pmatrix}
\]

invertible! sums over rows to 1

\[
T = \begin{pmatrix}
1.0 & 1.0 & 1.0 \\
1.0 & 1.0 & 1.0 \\
1.0 & 1.0 & 1.0
\end{pmatrix}
\]

* \(\frac{1}{3}\)

not invertible! sums over rows to 1

In observation space the augmentation function is random
e.g. \(f(1) = \{=0 \text{ with p=0.2}, =1 \text{ with p=0.6}, \text{ and } =2 \text{ with p=0.2}\}\)

deterministic linear operator in the distribution space

in observation space
f(x) is either \{0, 1, 2\}
with equal probabilities

not invertible!
Markov operator

We want $T(x)$ such that $T(P) = T(Q) \Leftrightarrow P = Q$, i.e. we want an invertible operator “$T$: distribution $\rightarrow$ distribution”.

Example: $T(P) = P \ast N(0, 1)$, i.e. $T(x) = x + \epsilon$, $\epsilon \sim N(0, 1)$, i.e. $T[P](x) = [P \ast N(0, 1)](x)$, $T^{-1}(T(P)) = P$, $T^{-1}(W) = \text{deconv}(W)$

This is like an infinite dimensional vector and the operator $T$ is also infinite dimensional.
How to test invertibility of an infinite-dimensional operator?

General statements:

1. A **composition** of invertible operators is **invertible** (i.e. a sequence of “good”/“non-leaking” augmentations is still good)

2. A **linear combination** of invertible operators is **not necessarily invertible**

invertible \([½(T_1 + T_2)](P)\) means randomly choosing between augmentations \(T_1\) and \(T_2\) and applying it to a single sample from \(P\)

\[
T_1[P](x) = P(x-1) \\
T_2[P](x) = P(x+1)
\]

\[
\frac{1}{2}(T_1(P) + T_2(P)) = [\frac{1}{2}(T_1 + T_2)](P)
\]

[“Training Generative Adversarial Networks with Limited Data”, Karras et al., NeurIPS’20]
How to test invertibility of an infinite-dimensional operator?

\[ T = \sum_{i=0}^{N-1} p_i G^i \]
\[ U = \sum_{j=0}^{N-1} q_j G^j \]
\[ UT = \left( \sum_{i=0}^{N-1} p_i G^i \right) \left( \sum_{j=0}^{N-1} q_j G^j \right) = \sum_{i,j=0}^{N-1} p_i q_j G^{i+j} \]

\[ = \sum_{k=0}^{N-1} \left[ p \otimes q \right]_k G^k \]
\[ = \sum_{k=0}^{N-1} \left[ F^{-1} \left( \hat{p} \otimes \hat{q}^{-1} \right) \right]_k G^k \]

if we set \[ \hat{q}_i = \frac{1}{\hat{p}_i} \]
\[ = \sum_{k=0}^{N-1} \left[ F^{-1} \left( \hat{p} \otimes \hat{p}^{-1} \right) \right]_k G^k = \sum_{k=0}^{N-1} \left[ F^{-1} 1 \right]_k G^k = G^0 = T \]

Inverse operator exists if there are no zeros in operator’s Fourier spectre.

Solution: uniform but with higher probability of \( G^0 \) like \([0.28 \ 0.24 \ 0.24 \ 0.24]\) - has no zeros in spectrum!
Essentially \( T = [(1-\alpha) \times \text{Uniform} + \alpha \times \text{Identity}] \) - e.g. almost like a regularization.

[“Training Generative Adversarial Networks with Limited Data”, Karras et al., NeurIPS’20]
How to test invertibility of an infinite-dimensional operator?

*Inverse operator exists if there are no zeros in operator’s Fourier spectre.*

**Solution:** uniform but with higher probability of $G^0$ like $[0.28, 0.24, 0.24, 0.24]$ - has no zeros in spectrum!

Essentially $T = [(1-\alpha) \cdot \text{Uniform} + \alpha \cdot \text{Identity}]$ - e.g. almost like a regularization.

[“Training Generative Adversarial Networks with Limited Data”, Karras et al., NeurIPS’20]
How to test invertibility of an infinite-dimensional operator?

Other cases:
1. Non-compact discrete groups (integer shift): also “non-zero Fourier”
2. For continuous groups (e.g. rotations): also “non-zero Fourier” (use Haar measure over that groups under the integral);
3. Additive pixel noise: “non-zero Fourier” of the noise kernel
4. Cropping / blitting / “projection”: requires $P(\text{identity}) > 0$

Assume $\exists y \neq z$ s.t. $Ty = Tz$, e.g. $T(y - z) = 0$, e.g. $Tx = 0$.

$$\mathcal{T} = p_0 \mathcal{I} + \sum_{j=1}^{N} p_j \mathcal{P}_j \quad 0 = \mathcal{T}x = p_0 x + \sum_{j=1}^{N} p_j \mathcal{P}_j x \quad \sum_{j=1}^{N} p_j \langle x, \mathcal{P}_j x \rangle = -p_0 \langle x, x \rangle \geq 0$$

invertible if $p_0 \neq 0$

[“Training Generative Adversarial Networks with Limited Data”, Karras et al., NeurIPS’20]
So how do we use it?

\[ \text{Q} \hspace{5em} \text{P} \]

\[ \mathbf{T} = \mathbf{T}_1 \circ \mathbf{T}_2 \circ \mathbf{T}_3 \ldots \]

\[ \mathbf{T}(\mathbf{Q}) \hspace{5em} \mathbf{T}(\mathbf{P}) \]

\[ (1-p)\mathbf{Q} + p\mathbf{T}(\mathbf{Q}) \]

\[ (1-p)\mathbf{P} + p\mathbf{T}(\mathbf{P}) \]

\[ D_{\text{acc}} = 0.9 \]

\[ D_{\text{acc}} = 0.55 \]

\[ D_{\text{acc}} = 0.8 \]

[“Training Generative Adversarial Networks with Limited Data”, Karras et al., NeurIPS’20]
Does it help? - yes!

\[ \text{real train} / \text{real val} / \text{generated} \]

\( D(x) \) scores trained on 20k samples

["Training Generative Adversarial Networks with Limited Data", Karras et al., NeurIPS'20]
Does it help? - yes!

[“Training Generative Adversarial Networks with Limited Data”, Karras et al., NeurIPS’20]
Takeaway

1. Regularizing the discriminator with augmentations helps.
2. But it has to be done in a way that does not “leak” into generated images.
3. For a wide variety of transformations, applying them with a fixed probability “does not leak” into generated examples.

Comments

1. All these methods still require careful parameter annealing.
2. As a result we can not reason about the convergence of an objective because there is no single objective! (we change it as we train)

[“Training Generative Adversarial Networks with Limited Data”, Karras et al., NeurIPS’20]
CycleGAN overview

\[ D_X \xrightarrow{G} Y \xleftarrow{F} X \quad D_Y \xrightarrow{G} Y \xleftarrow{F} X \]

(a) cycle-consistency loss

(b) cycle-consistency loss

(c) “identity loss”

(d) “identity loss” (applying \( Y \xrightarrow{F} X \) mapping to \( X \))

[“Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”, Zhu et al., ICCV’17]
How to reason about the complexity of the CycleGAN?

CycleGAN trained to map MNIST train split to the MNIST test split.

[“Kernel of CycleGAN as a Principle homogeneous space”, Moriakov et al., ICLR’20]
[“Guiding the One-to-One Mapping in CycleGAN via Optimal Transport’, Lu et al., AAAI’19]
How to bound the unsupervised alignment error?

\[
\text{prediction error} < \text{smallest unsupervised alignment error} + \text{smallest approximation error in H} + \text{the variance between functions minimizing the alignment loss.}
\]

[“Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”, Galanti et al., JMLR’21]
[“Estimating the Success of Unsupervised Image to Image Translation”, Benaim et al., ECCV’18]
[“The role of minimal complexity functions in unsupervised learning of semantic mappings”, Galanti et al., ICLR’18]
How to bound the unsupervised alignment error?

$$R_{D_A}[h_1, y] \leq \sup_{h_2 \in \mathcal{P}_\omega} R_{S_A}[h_1, h_2] + c \inf_{h \in \mathcal{P}_\omega} \rho_C(h \circ S_A, S_B) + \ldots$$

$R$ - pred error; $\rho$ - alignment error; $D_A$ - distribution; $S_A$ - dataset; $P_k$ - hypotheses with “low” alignment error

How to use this bound?

A given choice of hyperparameters is evaluated as follows:

1. $\inf_{h \in \mathcal{P}_k} \rho_C(h \circ S_A, S_B)$ - we minimize the “GAN loss” to get “the first best” $h_1$

2. $\min_{h_2 \in \mathcal{H}_k} \left\{ \rho_C(h_2 \circ S_A, S_B) - \lambda R_{S_A}[h_1, h_2] \right\}$ - then pick the “second best” $h_2$

3. $R_{S_A}[h_1, h_2] + \rho_C(h_1 \circ S_A, S_B)$ - and then bound the unknown GT error

[“Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”, Galanti et al., JMLR’21]
[“Estimating the Success of Unsupervised Image to Image Translation”, Benaim et al., ECCV’18]
[“The role of minimal complexity functions in unsupervised learning of semantic mappings”, Galanti et al., ICLR’18]
The ground truth error and the theoretical bound as a function of hyper-parameter optimization steps varying
- encoder and decoder layers
- batch size
- learning rate
...

[“Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”, Galanti et al., JMLR’21]
[“Estimating the Success of Unsupervised Image to Image Translation”, Benaim et al., ECCV’18]
[“The role of minimal complexity functions in unsupervised learning of semantic mappings”, Galanti et al., ICLR’18]
Theorem 1 (Cross-Domain Mapping with IPMs) Assume that $X_A \subset \mathbb{R}^N$ and $X_B \subset \mathbb{R}^M$ are convex and bounded sets. Let $\mathcal{H}$ be the hypothesis class and $\mathcal{C}$ the class of discriminators. Assume that $\mathcal{C} \subset C^2$ and $\sup_{d \in \mathcal{C}} \|d\|_\infty, X_A \cup X_B < \infty$. Then, for any $\delta \in (0, 1)$ and $c \geq 1$, with probability at least $1 - \delta$ over the selection of $S_A \sim D_A^{m_1}$ and $S_B \sim D_B^{m_2}$, for every $\omega \in \Omega$ and $h_1 \in \mathcal{P}_\omega := \mathcal{P}_\omega(S_A, S_B)$, we have:

- **Translation functions**
- **Discriminator class**
- **Domains**
- **Single discriminator**
- **Finite datasets**
- **Hyperparams**
- **A mapping produced by the learning algorithm**

[“Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”, Galanti et al., JMLR’21]
[“Estimating the Success of Unsupervised Image to Image Translation”, Benaim et al., ECCV’18]
[“The role of minimal complexity functions in unsupervised learning of semantic mappings”, Galanti et al., ICLR’18]
Theorem 1 (Cross-Domain Mapping with IPMs) Assume that $\mathcal{X}_A \subset \mathbb{R}^N$ and $\mathcal{X}_B \subset \mathbb{R}^M$ are convex and bounded sets. Let $\mathcal{H}$ be the hypothesis class and $\mathcal{C}$ the class of discriminators. Assume that $\mathcal{C} \subset C^2$ and $\sup_{d \in \mathcal{C}} \|d\|_{\infty, \mathcal{X}_A \cup \mathcal{X}_B} < \infty$. Then, for any $\delta \in (0, 1)$ and $c \geq 1$, with probability at least $1 - \delta$ over the selection of $S_A \sim D_A^{m_1}$ and $S_B \sim D_B^{m_2}$, for every $\omega \in \Omega$ and $h_1 \in \mathcal{P}_\omega := \mathcal{P}_\omega(S_A, S_B)$, we have:

$$R_{D_A}[h_1, y] \lesssim \sup_{h_2 \in \mathcal{P}_\omega} R_{S_A}[h_1, h_2] + c \inf_{h \in \mathcal{P}_\omega} \rho_{\mathcal{C}}(h \circ S_A, S_B) + \inf_{h \in \mathcal{P}_\omega} \inf_{d \in \mathcal{C}} \mathcal{K}(h, d; y)$$

$$+ \hat{\mathcal{R}}_{S_A}(e_H) + \hat{\mathcal{R}}_{S_A}(\mathcal{C} \circ \mathcal{H}) + \hat{\mathcal{R}}_{S_B}(\mathcal{C}) + \sqrt{\frac{\log(1/\delta)}{\min(m_1, m_2)}}$$

where, $\mathcal{K}(h, d; y) := \mathbb{E}_{x \sim D_A} \left[\|\nabla_y(x)d(y(x)) - (h(x) - y(x))\|_2\right]$. 

[“Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”, Galanti et al., JMLR’21]
[“Estimating the Success of Unsupervised Image to Image Translation”, Benaim et al., ECCV’18]
[“The role of minimal complexity functions in unsupervised learning of semantic mappings”, Galanti et al., ICLR’18]
Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs

Galanti et al., JMLR’21

Lem 3:

\[ R_{DA}[h_1, y] \leq 3 \sup_{h_2 \in \mathbb{P}} R_{SA}[h_1, h_2] + 3 \inf_{h \in \mathbb{P}} R_{DA}[h, y] \leq L \]

Lem 4:

\[ R_{DA}[h, y] \leq \frac{2 \rho_C(h \circ D_A, D_B)}{2 - \beta(d)} + \frac{2 \sup_{u \in X_A} \|h(u) - y(u)\|_2}{2 - \beta(d)} \cdot K(h, d; y) \]

Lem 7:

\[ R_{DA}[h_1, h_2] \leq R_{SA}[h_1, h_2] + 2 \hat{\mathcal{R}}_{SA}(\ell_H) + 9K^2 \sqrt{\frac{\log(6/\delta)}{2m_1}} \]

[“Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”, Galanti et al., JMLR’21]
Lemma 3 (Triangle inequality and the “set diameter”)

\[ R_{DA}[h_1, y] \leq 3 \sup_{h_2 \in \mathcal{P}} R_{DA}[h_1, h_2] + 3 \inf_{h \in \mathcal{P}} R_{DA}[h, y] \]

\[
\|a - c\|_2^2 \leq (\|a - b\|_2 + \|b - c\|_2)^2 \\
\leq \|a - b\|_2^2 + \|b - c\|_2^2 + 2 \max(\|a - b\|_2^2, \|b - c\|_2^2) \\
\leq 3(\|a - b\|_2^2 + \|b - c\|_2^2)
\]

\[
R_{DA}[h_1, y] = \mathbb{E}_{x \sim DA}[\|h_1(x) - y(x)\|_2^2] \\
\leq \mathbb{E}_{x \sim DA} \left[ 3\|h_1(x) - h^*(x)\|_2^2 + 3\|h^*(x) - y(x)\|_2^2 \right] \\
= 3 \left[ R_{DA}[h_1, h^*] + \inf_{h \in \mathcal{P}} R_{DA}[h, y] \right]
\]

\[
R_{DA}[h_1, h^*] \leq \sup_{h_2 \in \mathcal{P}} R_{DA}[h_1, h_2]
\]
**Lem 4 (Prediction Error via Stat. Distance and Discriminator Capacity)**

\[
\rho_C(h \circ D_A, D_B) = \sup_{d \in C} \left\{ \mathbb{E}_{u \sim h \circ D_A} [d(u)] - \mathbb{E}_{v \sim D_B} [d(v)] \right\} = \sup_{d \in C} \left\{ \mathbb{E}_{x \sim D_A} [d \circ h(x) - d \circ y(x)] \right\}
\]

\[
\geq \mathbb{E}_{x \sim D_A} [d(h(x)) - d(y(x))] = \mathbb{E}_{x \sim D_A} \left[ \|h(x) - y(x)\|_2^2 \right]
\]

\[= \sup_{u \in X A} \|h(u) - y(u)\|_2 \cdot \mathbb{E}_{x \sim D_A} \left[ \|\nabla y(x)d(y(x)) - (h(x) - y(x))\|_2 \right]
\]

\[= \left(1 - \frac{\beta(d)}{2}\right) R_{DA}[h, y] - \sup_{u \in X A} \|h(u) - y(u)\|_2 \cdot \mathbb{K}(h, d; y)
\]

\[
\text{Prediction error} \quad \text{avg prediction error} \quad \text{stat. distance} \quad \text{max prediction error < } L \quad \text{discriminator error}
\]

\[
R_{DA}[h, y] \leq \frac{2\rho_C(h \circ D_A, D_B)}{2 - \beta(d)} + \frac{2}{2 - \beta(d)} \cdot \mathbb{K}(h, d; y)
\]

\[
\text{where, } \mathbb{K}(h, d; y) := \mathbb{E}_{x \sim D_A} \left[ \|\nabla y(x)d(y(x)) - (h(x) - y(x))\|_2 \right]
\]

\[
f(x) - f(y) = <\nabla f(x), x - y> + ... = \|x - y\|^2 + <\nabla f(x) - (x - y), x - y> + ...
\]

[“Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”, Galanti et al., JMLR’21]
Definition 3.1 (Empirical Rademacher complexity)

\[ \hat{\mathcal{R}}_S(\mathcal{G}) = \mathbb{E}_\sigma \left[ \sup_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i) \right] \]

Theorem 3.3 Let \( \mathcal{G} \) be a family of functions mapping from \( \mathcal{Z} \) to \([0, 1]\). With probability at least \( 1 - \delta \)

\[ \mathbb{E}[g(z)] \leq \frac{1}{m} \sum_{i=1}^{m} g(z_i) + 2\hat{\mathcal{R}}_S(\mathcal{G}) + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}} \]

\[ \mathbb{E}[\Phi(S)] = \mathbb{E}_S \left[ \sup_{g \in \mathcal{G}} \left( \mathbb{E}[g] - \hat{\mathbb{E}}_S(g) \right) \right] = \mathbb{E}_S, S', \sigma \left[ \sup_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i (g(z'_i) - g(z_i)) \right] \]

\[ \leq \mathbb{E}_{\sigma, S', \sigma} \left[ \sup_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z'_i) \right] + \mathbb{E}_{\sigma, S} \left[ \sup_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} -\sigma_i g(z_i) \right] \]

\[ = 2 \mathbb{E}_{\sigma, S} \left[ \sup_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i) \right] = 2\mathcal{R}_m(\mathcal{G}). \]

with probability at least \( 1 - \delta \)

by McDiarmid’s inequality \[ \Phi(S) \leq \mathbb{E}_S[\Phi(S)] + \sqrt{\frac{\log \frac{2}{\delta}}{2m}} \quad \text{and} \quad \mathcal{R}_m(\mathcal{G}) \leq \hat{\mathcal{R}}_S(\mathcal{G}) + \sqrt{\frac{\log \frac{2}{\delta}}{2m}} \]

Lem 7 (Sample complexity)

\[ \mathbb{E}[g(z)] \leq \frac{1}{m} \sum_{i=1}^{m} g(z_i) + 2\hat{R}_S(G) + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}} \]

\[ R_{DA}[h_1, h_2] \leq R_{SA}[h_1, h_2] + 2\hat{R}_S(\ell_H) + 9K^2 \sqrt{\frac{\log(6/\delta)}{2m_1}} \]

\[ \rho_C(D_B, S_B) = \sup_{d \in \mathcal{C}} \left\{ \mathbb{E}_{x \sim D_B}[d(x)] - \frac{1}{m_2} \sum_{x \in S_B} d(x) \right\} \preceq \hat{R}_S(C) + \sqrt{\frac{\log(1/\delta)}{m_2}} \]

\[ \rho_C(h \circ D_A, h \circ S_A) \preceq \hat{R}_S(C \circ H) + \sqrt{\frac{\log(1/\delta)}{m_1}} \]
Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”, Galanti et al., JMLR’21

Lem 3: 
\[ R_{DA}[h_1, y] \leq 3 \sup_{h_2 \in \mathcal{P}} R_{DA}[h_1, h_2] + 3 \inf_{d \leq 1} K(h, d; y) \]

Lem 4: 
\[ R_{DA}[h, y] \leq \frac{2 \rho_C(h \circ D_A, D_B)}{2 - \beta(d)} + \frac{2 \sup_{u \in X_A} \| h(u) - y(u) \|_2}{2 - \beta(d)} \cdot K(h, d; y) \]

Lem 7: 
\[ R_{DA}[h_1, h_2] \leq R_{SA}[h_1, h_2] + 9 K^2 \sqrt{\frac{\log(6/\delta)}{2m_1}} \]

\[ R_{DA}[h_1, y] \leq \sup_{h_2 \in \mathcal{P}_w} R_{SA}[h_1, h_2] + c \inf_{h \in \mathcal{P}_w} \rho_C(h \circ S_A, S_B) + \inf_{h \in \mathcal{P}_w} \inf_{d \leq 1} K(h, d; y) \]

\[ + \hat{R}_{SA}(\ell_H) + \hat{R}_{SA}(C \circ \mathcal{H}) + \hat{R}_{SB}(C) \]

\[ + \sqrt{\frac{\log(1/\delta)}{\min(m_1, m_2)}} \]
Takeaway

The **prediction error** of the **unsupervised** image translation method (wrt the ground truth output) can be bounded via
- **minimal statistical distance** attainable by the network and
- **variance between solutions** that attain that lowest statistical distance.

And this bound actually works in practice!

Comments

1. Regression CNNs can fit almost random (x,y) pairs - can I2I networks fit random (x1, x2) pairs? if so why doesn’t R[h1, h2] explode?
2. The empirical **Rademacher complexity** of the discriminator class seems related to the “**expected statistical distance between random splits** of that dataset”?
Recap

1. “Neural GAN distances” between datasets seem to have have **better sample complexity** than “classical” distances. We used an $\epsilon$-net over NN weights and Chernoff bound on each element of $\epsilon$-net to show that.

2. These neural distances can be “smoothened” via **instance noise and augmentations** to make gradient descent iterations more stable. By treating random augmentations as Markov operators we showed that in most cases **skipping augmentations with fixed probability** ensures that the neural distance remain “non-leaking” even under augmentations.

3. The **prediction error** of the unsupervised alignment method can be bounded via the **variance between solutions** attaining similar GAN loss.
Thank you for your time!

Main papers:

