Toward Understanding Generative Data Augmentation

Chenyu Zheng (HeXuan)[1] Guoqiang Wu[2] Chongxuan Li[1]

[1]Gaoling School of AI, Renmin University of China

[2]School of Software, Shandong University

NeurIPS 2023

Table of Contents

Table of Contents

1.1 Generative data augmentation (GDA)

Generative data augmentation, which scales datasets by generating labeled examples from a trained conditional generative model, boosts classification performance in various tasks.

Figure: Example 1024×1024 images from the Imagen model.

1.2 GDA helps supervised learning

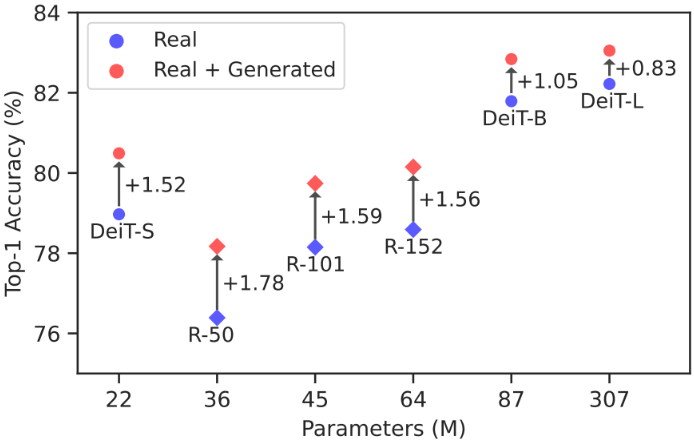


Figure: Comparison of classifier performance when 1.2M generated images are used for GDA [1].

1.3 GDA helps semi-supervised learning

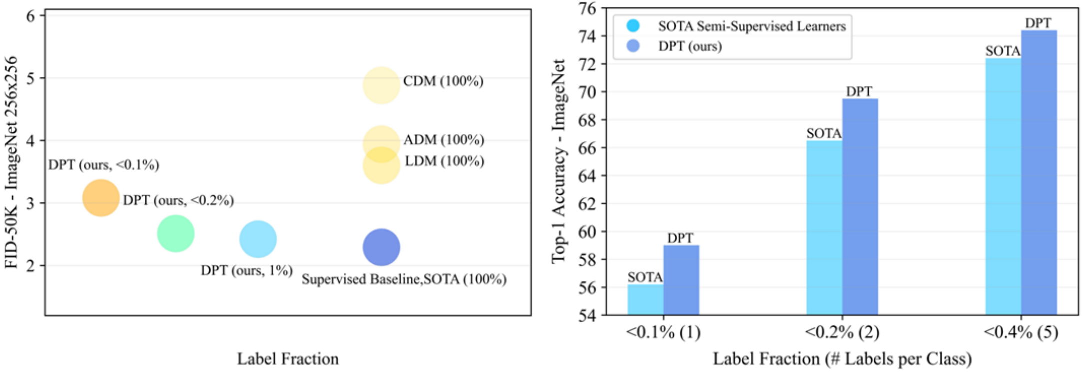


Figure: DPT improves the state-of-the-art semi-supervised learner [2].

1.4 GDA helps adversarial learning

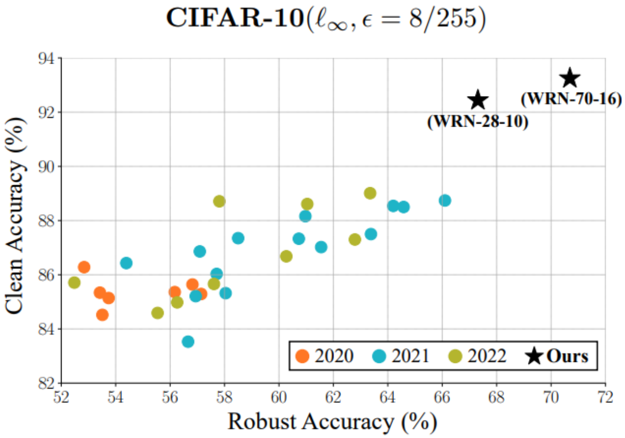


Figure: GDA improves the robustness of deep models [3].

1.5 Open problem

Lack of theoretical understanding

Little work has investigated the GDA from a theoretical perspective.

Our contributions

- We establish a general theoretical framework for the GDA in the supervised learning setting.
- We particularize the general results to the binary Gaussian mixture model (bGMM) and generative adversarial nets (GANs).
- We conduct experiments to validate our theoretical findings.

Table of Contents

2.1 Notions and definitions

- **Data:**
  - Let $\mathcal{X} \subseteq \mathbb{R}^n$ be the input space and $\mathcal{Y}$ be the label space.
  - We denote by $\mathcal{D}$ the population distribution over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$.

2.1 Notions and definitions

- **Data:**
  - Let $\mathcal{X} \subseteq \mathbb{R}^n$ be the input space and $\mathcal{Y}$ be the label space.
  - We denote by $\mathcal{D}$ the population distribution over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$.
- **Learning algorithm:**
  - Let $\mathcal{A}$ be a learning algorithm.
  - Let $\mathcal{A}(S) \in (\mathcal{Y})^{\mathcal{X}}$ be the hypothesis learned on the dataset S.

2.1 Notions and definitions

- **Data:**
  - Let $\mathcal{X} \subseteq \mathbb{R}^n$ be the input space and $\mathcal{Y}$ be the label space.
  - We denote by $\mathcal{D}$ the population distribution over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$.
- **Learning algorithm:**
  - Let $\mathcal{A}$ be a learning algorithm.
  - Let $\mathcal{A}(S) \in (\mathcal{Y})^{\mathcal{X}}$ be the hypothesis learned on the dataset S.
- **Evaluation:**
  - Loss function $\ell : (\mathcal{Y})^{\mathcal{X}} \times \mathcal{Z} \to \mathbb{R}_+$.
  - True error $\mathcal{R}_{\mathcal{D}}(\mathcal{A}(S))$ with respect to the data distribution $\mathcal{D}$ is defined as $\mathbb{E}_{\mathbf{z}\sim\mathcal{D}}[\ell(\mathcal{A}(S), \mathbf{z})]$.
  - Empirical error $\widehat{\mathcal{R}}_S(\mathcal{A}(S))$ is defined as $\frac{1}{m}\sum_{i=1}^{m} \ell(\mathcal{A}(S), \mathbf{z}_i)$.

2.1 Notions and definitions

- **Training generative model:** given a dataset S with m_S i.i.d. examples from $\mathcal{D}$, we can train a conditional generative model G with the model distribution $\mathcal{D}_G(S)$.

2.1 Notions and definitions

- **Training generative model:** given a dataset S with m_S i.i.d. examples from $\mathcal{D}$, we can train a conditional generative model G with the model distribution $\mathcal{D}_G(S)$.
- **Generative data augmentation:** we then obtain a new dataset S_G with m_G i.i.d. samples from $\mathcal{D}_G(S)$, where m_G is a hyperparameter.

2.1 Notions and definitions

- **Training generative model:** given a dataset S with m_S i.i.d. examples from $\mathcal{D}$, we can train a conditional generative model G with the model distribution $\mathcal{D}_G(S)$.
- **Generative data augmentation:** we then obtain a new dataset S_G with m_G i.i.d. samples from $\mathcal{D}_G(S)$, where m_G is a hyperparameter.
- We denote the total number of the data in augmented set $\widetilde{S} = S \cup S_G$ by m_T.
- We define the mixed distribution after augmentation as $\widetilde{\mathcal{D}}(S) = \frac{m_S}{m_T}\mathcal{D} + \frac{m_G}{m_T}\mathcal{D}_G(S)$

2.1 Notions and definitions

- **Training generative model:** given a dataset S with m_S i.i.d. examples from $\mathcal{D}$, we can train a conditional generative model G with the model distribution $\mathcal{D}_G(S)$.
- **Generative data augmentation:** we then obtain a new dataset S_G with m_G i.i.d. samples from $\mathcal{D}_G(S)$, where m_G is a hyperparameter.
- We denote the total number of the data in augmented set $\widetilde{S} = S \cup S_G$ by m_T.
- We define the mixed distribution after augmentation as $\widetilde{\mathcal{D}}(S) = \frac{m_S}{m_T}\mathcal{D} + \frac{m_G}{m_T}\mathcal{D}_G(S)$

Our goal

We interested in the generalization error *Gen-error* $= |\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\widetilde{S})) - \widehat{\mathcal{R}}_{\widetilde{S}}(\mathcal{A}(\widetilde{S}))|$. We will derive a high probability bound for it by using the algorithmic stability technique.

2.2 Algorithmic stability

Algorithmic stability analysis is an important tool to provide generalization guarantees, which exploits particular properties of the algorithm and provides algorithm-dependent bound.
Given a set $S = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m\}$, we define S^i as the set after replacing the i-th data point with $\mathbf{z}_i'$ in the set S.

Definition 1 (Uniform stability)

Algorithm $\mathcal{A}$ is uniformly β_m-stable with respect to the loss function ℓ if the following holds

$$\forall S \in \mathcal{Z}^m, \forall \mathbf{z} \in \mathcal{Z}, \forall i \in [m], \sup_{\mathbf{z}} \left| \ell(\mathcal{A}(S), \mathbf{z}) - \ell(\mathcal{A}(S^i), \mathbf{z}) \right| \leq \beta_m.$$

Understanding the stability

Intuitively, the more stable an algorithm is, the less sensitive it is to the input, and thus less likely to overfit.

2.3 Stability bound in the i.i.d. setting

[4] proposed a moment bound and obtained a nearly optimal generalization guarantee, which only requires $\beta_m = o(1/\log m)$ to converge.

Theorem 2 (Corollary 8, [4])

Assume that $\mathcal{A}$ is a β_m-stable learning algorithm and the loss function ℓ is bounded by M. Given a training set S with m i.i.d. examples sampled from the distribution $\mathcal{D}$, then for any $\delta \in (0,1)$, with probability at least $1-\delta$, it holds that

$$\left|\mathcal{R}_{\mathcal{D}}(\mathcal{A}(S)) - \widehat{\mathcal{R}}_S(\mathcal{A}(S))\right| \lesssim \log(m)\beta_m \log\left(\frac{1}{\delta}\right) + M\sqrt{\frac{1}{m}\log\left(\frac{1}{\delta}\right)}.$$

2.4 GDA is a non-i.i.d setting

Mismatch with the classical results

GDA is a non-i.i.d setting:

- The distribution $\mathcal{D}_G(S)$ learned by the generative model is generally not the same as the true distribution $\mathcal{D}$.
- The learned model distribution $\mathcal{D}_G(S)$ is heavily dependent on the sampled dataset S.

2.4 GDA is a non-i.i.d setting

Mismatch with the classical results

GDA is a non-i.i.d setting:

- The distribution $\mathcal{D}_G(S)$ learned by the generative model is generally not the same as the true distribution $\mathcal{D}$.
- The learned model distribution $\mathcal{D}_G(S)$ is heavily dependent on the sampled dataset S.

First attempt

We try to use the existing non-i.i.d stability bounds [5].

2.5 Stability bounds for mixing processes

Existing stability bounds for mixing processes only focus on the stationary sequence.

Definition 3 (Stationary sequence)

A sequence of random variables $\mathbf{Z} = \{Z_t\}_{t=-\infty}^{\infty}$ is said to be stationary if for any t and non-negative integers m and k, the random vectors $(Z_t, \ldots, Z_{t+m})$ and $(Z_{t+k}, \ldots, Z_{t+m+k})$ have the same distribution.

2.5 Stability bounds for mixing processes

Existing stability bounds for mixing processes only focus on the stationary sequence.

Definition 3 (Stationary sequence)

A sequence of random variables $\mathbf{Z} = \{Z_t\}_{t=-\infty}^{\infty}$ is said to be stationary if for any t and non-negative integers m and k, the random vectors $(Z_t, \ldots, Z_{t+m})$ and $(Z_{t+k}, \ldots, Z_{t+m+k})$ have the same distribution.

Problem

Unfortunately, the GDA setting in this paper does not satisfy the stationary condition, because $(\mathbf{z}_1, \ldots, \mathbf{z}_{m_S}) = S$ and $(\mathbf{z}_{m_S+1}, \ldots, \mathbf{z}_{2m_S}) \subseteq S_G$ do not have the same distribution.

2.6 Stability bounds for dependence graph

The dependence graph reflects the dependence between random variables.

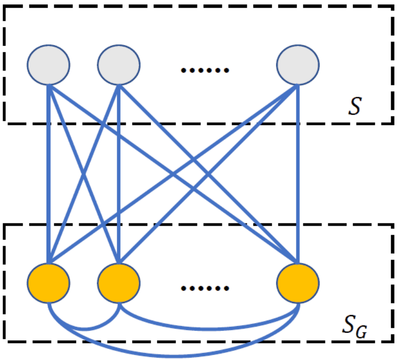


Figure: Dependence graph in the GDA setting.

2.6 Stability bounds for dependence graph

Theorem 4

Assume that $\mathcal{A}$ is a β_m-stable. Given a set $\widetilde{S}$ of size m sampled from the same marginal distribution $\mathcal{D}$ with dependency graph G. Suppose the maximum degree of G is Δ, and the loss function ℓ is bounded by M. For any $\delta \in (0,1)$, with probability at least $1-\delta$, it holds that

$$\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\widetilde{S})) \leq \widehat{\mathcal{R}}_{\widetilde{S}}(\mathcal{A}(\widetilde{S})) + 2\beta_{m,\Delta}(\Delta+1) + (4\beta_m + \frac{M}{m})\sqrt{\frac{\Lambda(G)}{2}\log(\frac{1}{\delta})},$$

where $\beta_{m,\Delta} = \max_{i \leq \Delta} \beta_{m-i}$ and $\Lambda(G)$ is the forest complexity of the dependence graph G.

2.6 Stability bounds for dependence graph

Problems

- Theorem 4 requires $\widetilde{S}$ sampled from the same marginal distribution $\mathcal{D}$, which fails to hold in the context of GDA.
- When $m_G = 0$ and $\widetilde{S} = S$, Theorem 4 requires $\beta_m = o(1/\sqrt{m})$ to converge.
- Theorem 4 is proposed for the general case with data dependence and **does not consider the property of special cases**. In the case of strong dependence like GDA, the **forest complexity may be too large to give a meaningful bound**:

$$\Lambda(G) \leq m_S(1 + m_G)^2 + 1^2 \lesssim m_S m_G^2,$$

$$\frac{M}{m_T}\sqrt{\frac{\Lambda(G)}{2}\log(\frac{1}{\delta})} \lesssim \frac{M}{m_T}\sqrt{\frac{m_S m_G^2}{2}\log(\frac{1}{\delta})} \lesssim M\sqrt{\frac{m_S}{2}\log(\frac{1}{\delta})},$$

which fails to converge.

Table of Contents

3.1 Proof idea

Recall that $\widetilde{\mathcal{D}}(S)$ has been defined as the mixed distribution, we first decomposed *Gen-error* as

$$\begin{aligned} |\textit{Gen-error}| &= |\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\widetilde{S})) - \widehat{\mathcal{R}}_{\widetilde{S}}(\mathcal{A}(\widetilde{S}))| \\ &\leq \underbrace{\left|\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\widetilde{S})) - \mathcal{R}_{\widetilde{\mathcal{D}}(S)}(\mathcal{A}(\widetilde{S}))\right|}_{\text{Distributions' divergence}} + \underbrace{\left|\mathcal{R}_{\widetilde{\mathcal{D}}(S)}(\mathcal{A}(\widetilde{S})) - \widehat{\mathcal{R}}_{\widetilde{S}}(\mathcal{A}(\widetilde{S}))\right|}_{\text{Generaliztion error w.r.t. mixed distribution}} . \end{aligned}$$

3.1 Proof idea

Recall that $\widetilde{\mathcal{D}}(S)$ has been defined as the mixed distribution, we first decomposed *Gen-error* as

$$\begin{aligned}|\textit{Gen-error}| &= |\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\widetilde{S})) - \widehat{\mathcal{R}}_{\widetilde{S}}(\mathcal{A}(\widetilde{S}))| \\ &\leq \underbrace{\left|\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\widetilde{S})) - \mathcal{R}_{\widetilde{\mathcal{D}}(S)}(\mathcal{A}(\widetilde{S}))\right|}_{\text{Distributions' divergence}} + \underbrace{\left|\mathcal{R}_{\widetilde{\mathcal{D}}(S)}(\mathcal{A}(\widetilde{S})) - \widehat{\mathcal{R}}_{\widetilde{S}}(\mathcal{A}(\widetilde{S}))\right|}_{\text{Generaliztion error w.r.t. mixed distribution}} .\end{aligned}$$

Main idea

- The first term can be bounded by the divergence (e.g., $\mathcal{D}_{\mathrm{TV}}, \mathcal{D}_{\mathrm{KL}}$) between the mixed distribution $\widetilde{\mathcal{D}}(S)$ and the true distribution $\mathcal{D}$. It is heavily dependent on the ability of the chosen generative model.

3.1 Proof idea

Recall that $\widetilde{\mathcal{D}}(S)$ has been defined as the mixed distribution, we first decomposed *Gen-error* as

$$\begin{aligned}|\textit{Gen-error}| &= |\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\widetilde{S})) - \widehat{\mathcal{R}}_{\widetilde{S}}(\mathcal{A}(\widetilde{S}))| \\ &\leq \underbrace{\left|\mathcal{R}_{\mathcal{D}}(\mathcal{A}(\widetilde{S})) - \mathcal{R}_{\widetilde{\mathcal{D}}(S)}(\mathcal{A}(\widetilde{S}))\right|}_{\text{Distributions' divergence}} + \underbrace{\left|\mathcal{R}_{\widetilde{\mathcal{D}}(S)}(\mathcal{A}(\widetilde{S})) - \widehat{\mathcal{R}}_{\widetilde{S}}(\mathcal{A}(\widetilde{S}))\right|}_{\text{Generaliztion error w.r.t. mixed distribution}} .\end{aligned}$$

Main idea

- The first term can be bounded by the divergence (e.g., $\mathcal{D}_{\mathrm{TV}}, \mathcal{D}_{\mathrm{KL}}$) between the mixed distribution $\widetilde{\mathcal{D}}(S)$ and the true distribution $\mathcal{D}$. It is heavily dependent on the ability of the chosen generative model.
- For the second term, We mainly use a core property that S satisfies the i.i.d. assumption, and S_G satisfies the conditional i.i.d. assumption when S is fixed. Inspired by this property, we furthermore decompose this term to obtain an upper bound.

3.2 Decomposition of the second term

For function $f(S)$, we denote its L_p norm and conditional L_p norm with respect to S_V by $\|f\|_p = (\mathbb{E}[\,\|f\|^p])^{\frac{1}{p}}$ and $\|f\|_p\,(S_V) = (\mathbb{E}[\,\|f\|^p \mid S_V])^{\frac{1}{p}}$, respectively.

$$
\begin{aligned}
&\left\| m_T \left(\mathcal{R}_{\widetilde{\mathcal{D}}(S)}(\mathcal{A}(\widetilde{S})) - \widehat{\mathcal{R}}_{\widetilde{S}}(\mathcal{A}(\widetilde{S})) \right) \right\|_p \\
=& \left\| m_S \mathcal{R}_{\mathcal{D}}(\mathcal{A}(\widetilde{S})) + m_G \mathcal{R}_{\mathcal{D}_G(S)}(\mathcal{A}(\widetilde{S})) - \sum_{\mathbf{z}_i \in S} \ell(\mathcal{A}(\widetilde{S}), \mathbf{z}_i) - \sum_{\mathbf{z}_i \in S_G} \ell(\mathcal{A}(\widetilde{S}), \mathbf{z}_i) \right\|_p \\
\leq& \underbrace{\left\| m_S \mathcal{R}_{\mathcal{D}}(\mathcal{A}(\widetilde{S})) - \sum_{i=1}^{m_S} \ell(\mathcal{A}(\widetilde{S}), \mathbf{z}_i) \right\|_p}_{\|\Phi_1(S,S_G)\|_p} + \underbrace{\left\| m_G \mathcal{R}_{\mathcal{D}_G(S)}(\mathcal{A}(\widetilde{S})) - \sum_{i=1}^{m_G} \ell(\mathcal{A}(\widetilde{S}), \mathbf{z}_i^G) \right\|_p}_{\|\Phi_2(S,S_G)\|_p} \\
\leq& \left\| \Phi_1 - \mathbb{E}_{S_G \sim \mathcal{D}_G^{m_G}(S)} \Phi_1 \right\|_p + \left\| \mathbb{E}_{S_G \sim \mathcal{D}_G^{m_G}(S)} \Phi_1 \right\|_p + \sup_S \|\Phi_2\|_p\,(S).
\end{aligned}
$$

3.3 General theoretical result

Theorem 5 (Generalization bound for GDA)

Assume that $\mathcal{A}$ is a β_m-stable learning algorithm and the loss function ℓ is bounded by M. Given an augmented set $\widetilde{S}$, then for any $\delta \in (0,1)$, with probability at least $1-\delta$, it holds that

$$|\textit{Gen-error}| \lesssim \underbrace{\frac{m_G}{m_T} M \mathcal{D}_{\mathrm{TV}}\left(\mathcal{D}, \mathcal{D}_G(S)\right)}_{\textit{Distributions' divergence}} + \frac{M(\sqrt{m_S}+\sqrt{m_G}) + m_S\sqrt{m_G}\beta_{m_T}}{m_T}\sqrt{\log\left(\frac{1}{\delta}\right)}$$

$$+ \frac{\beta_{m_T}\left(m_S \log m_S + m_G \log m_G\right) + m_S \log m_S M \mathcal{T}(m_S, m_G)}{m_T} \log\left(\frac{1}{\delta}\right),$$

where $\mathcal{T}(m_S, m_G) = \sup_i \mathcal{D}_{\mathrm{TV}}\left(\mathcal{D}_G^{m_G}(S), \mathcal{D}_G^{m_G}(S^i)\right)$.

3.3 General theoretical result

We consider the order of the learning guarantee with respect to m_S here.

Remark (Selection of augmentation size)

An efficient augmentation size $m^*_{G,\text{order}}$ with regard to the order of m_S can be defined as:

$$m^*_{G,\text{order}} = \inf_{m_G} \{\text{generalization error w.r.t. mixed distribution} \lesssim \text{distributions' divergence}\}.$$

3.3 General theoretical result

We consider the order of the learning guarantee with respect to m_S here.

Remark (Selection of augmentation size)

An efficient augmentation size $m^*_{G,\text{order}}$ with regard to the order of m_S can be defined as:

$$m^*_{G,\text{order}} = \inf_{m_G} \{\text{generalization error w.r.t. mixed distribution} \lesssim \text{distributions' divergence}\}.$$

Corollary 6 (Sufficient conditions for GDA with (no) faster learning rate)

Assume the assumptions in Theorem 5 hold, then

- *if* $\mathcal{D}_{\text{TV}}(\mathfrak{D}, \mathfrak{D}_G(S)) = o\left(\max\left(\log(m)\beta_m, 1/\sqrt{m}\right)\right)$, *then GDA enjoys a faster learning rate.*
- *if* $\mathcal{D}_{\text{TV}}(\mathfrak{D}, \mathfrak{D}_G(S)) = \Omega\left(\max\left(\log(m)\beta_m, 1/\sqrt{m}\right)\right)$, *then GDA can not enjoy a faster learning rate.*

3.3 General theoretical result

Our result also shows the importance of the "stability" of the generative model training.

Remark (Stability of the learned distribution)

$\mathcal{T}(m_S, m_G) = \sup_i \mathcal{D}_{\mathrm{TV}}\left(\mathcal{D}_G^{m_G}(S), \mathcal{D}_G^{m_G}(S^i)\right)$ in Theorem 5 reflects the stability of the learned distribution. Our bound suggests that the more stable the model distribution is, the better generalization can be achieved by GDA.

3.3 General theoretical result

Our result also shows the importance of the "stability" of the generative model training.

Remark (Stability of the learned distribution)

$\mathcal{T}(m_S, m_G) = \sup_i \mathcal{D}_{\mathrm{TV}}\left(\mathcal{D}_G^{m_G}(S), \mathcal{D}_G^{m_G}(S^i)\right)$ in Theorem 5 reflects the stability of the learned distribution. Our bound suggests that the more stable the model distribution is, the better generalization can be achieved by GDA.

We can particularize our general theory to concrete settings.

Remark (Applied to specified settings)

To analyze specified GDA settings, we need to estimate terms M, β_{m_T}, $\mathcal{D}_{\mathrm{TV}}\left(\mathcal{D}, \mathcal{D}_G(S)\right)$ and $\mathcal{T}(m_S, m_G)$ in Theorem 5.

Table of Contents

4.1 bGMM setting

- **Distribution**: $y \sim \text{uniform}\{-1, 1\}$ and $\mathbf{x} \mid y \sim \mathcal{N}(y\boldsymbol{\mu}, \sigma^2 I_d)$, where $\|\boldsymbol{\mu}\|_2 = 1$ and $\sigma^2 > 0$.

4.1 bGMM setting

- **Distribution**: $y \sim \text{uniform}\{-1, 1\}$ and $\mathbf{x} \mid y \sim \mathcal{N}(y\boldsymbol{\mu}, \sigma^2 I_d)$, where $\|\boldsymbol{\mu}\|_2 = 1$ and $\sigma^2 > 0$.
- **Linear classifier**: $\widehat{y} = \text{sign}(\boldsymbol{\theta}^\top \mathbf{x})$. Given m samples, $\boldsymbol{\theta}$ is learned by minimizing the NLL loss:

$$l(\boldsymbol{\theta}, (\mathbf{x}, y)) = \frac{1}{2\sigma^2}(\mathbf{x} - y\boldsymbol{\theta})^\top (\mathbf{x} - y\boldsymbol{\theta}).$$

As a result, this learning algorithm will return $\widehat{\boldsymbol{\theta}} = \frac{1}{m}\sum_{i=1}^{m} y_i \mathbf{x}_i$, which satisfies $\mathbb{E}[\widehat{\boldsymbol{\theta}}] = \boldsymbol{\mu}$.

4.1 bGMM setting

- **Distribution**: $y \sim \text{uniform}\{-1, 1\}$ and $\mathbf{x} \mid y \sim \mathcal{N}(y\boldsymbol{\mu}, \sigma^2 I_d)$, where $\|\boldsymbol{\mu}\|_2 = 1$ and $\sigma^2 > 0$.
- **Linear classifier**: $\widehat{y} = \text{sign}(\boldsymbol{\theta}^\top \mathbf{x})$. Given m samples, $\boldsymbol{\theta}$ is learned by minimizing the NLL loss:

$$l(\boldsymbol{\theta}, (\mathbf{x}, y)) = \frac{1}{2\sigma^2}(\mathbf{x} - y\boldsymbol{\theta})^\top (\mathbf{x} - y\boldsymbol{\theta}).$$

  As a result, this learning algorithm will return $\widehat{\boldsymbol{\theta}} = \frac{1}{m}\sum_{i=1}^m y_i \mathbf{x}_i$, which satisfies $\mathbb{E}[\widehat{\boldsymbol{\theta}}] = \boldsymbol{\mu}$.
- **Generative model**: given m data points, let m_y be the number of samples in class y,

$$\widehat{\boldsymbol{\mu}}_y = \frac{\sum_{y_i = y} \mathbf{x}_i}{m_y}, \quad \widehat{\sigma}_k^2 = \sum_y \frac{m_y}{m} \frac{\sum_{y_i = y}(x_{ik} - \widehat{\mu}_{yk})^2}{m_y - 1},$$

  Based on the learned parameters, we can perform GDA by generating new samples from the distribution $y \sim \text{uniform}\{-1, 1\}$, $\mathbf{x} \mid y \sim \mathcal{N}(\widehat{\boldsymbol{\mu}}_y, \widehat{\Sigma})$, where $\widehat{\Sigma} = \text{diag}(\widehat{\sigma}_1^2, \ldots, \widehat{\sigma}_d^2)$.

4.2 Theoretical result

Theorem 7 (Generalization bound for bGMM)

Given a set S with m_S i.i.d. samples from the bGMM distribution $\mathcal{D}$ and an augmented set S_G with m_G i.i.d. samples drawn from the learned Gaussian mixture distribution, then with high probability at least $1-\delta$, it holds that

$$|\textit{Gen-error}| \lesssim \begin{cases} \frac{\log(m_S)}{\sqrt{m_S}} & \text{if fix } d \text{ and } m_G = 0, \\ \frac{\log^2(m_S)}{\sqrt{m_S}} & \text{if fix } d \text{ and } m_G = \Theta(m_S), \\ \frac{\log(m_S)}{\sqrt{m_S}} & \text{if fix } d \text{ and } m_G = m^*_{G,\mathrm{order}}, \\ d & \text{if fix } m_S. \end{cases}$$

Negative learning rate of GDA

Even though we estimate the sufficient statistics of the Gaussian mixture distribution directly, we can not enjoy a better learning rate.

4.2 Theoretical result

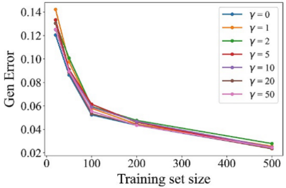
Negative learning rate of GDA

Even though we estimate the sufficient statistics of the Gaussian mixture distribution directly, we can not enjoy a better learning rate.

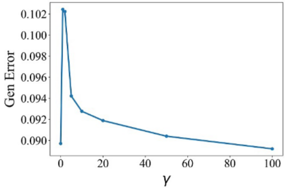
Improvement at a constant level matters a lot when overfitting happens

When m_S is small and d is large, the generalization error is awful. In this case, though GDA can only improve it at a constant level, the effect is obvious due to the large scale of d.

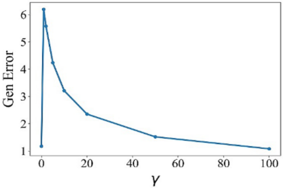
4.3 Simulation results



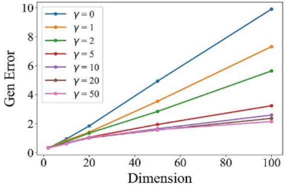
(a) $d = 1$, truth



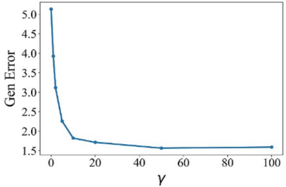
(b) $(d, m_S) = (1, 40)$, truth



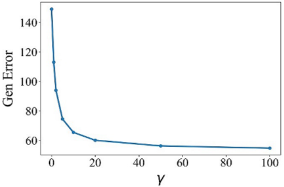
(c) $(d, m_S) = (1, 40)$, prediction



(d) $m_S = 10$, truth



(e) $(d, m_S) = (50, 10)$, truth



(f) $(d, m_S) = (50, 10)$, prediction

Figure: Simulations results with $\boldsymbol{\mu} = (1/\sqrt{d}, \dots, 1/\sqrt{d})^\top$ and $\sigma^2 = 0.6^2$

Table of Contents

5.1 GAN setting

- **Distribution**: $\mathcal{X} \subseteq [0,1]^d$ and $\mathcal{Y} = \{-1, 1\}$.

5.1 GAN setting

- **Distribution**: $\mathcal{X} \subseteq [0,1]^d$ and $\mathcal{Y} = \{-1, 1\}$.
- **Deep neural classifier**: L-layer MLP or CNN $f(\mathbf{w}, \cdot) : \mathcal{Z} \to \mathbb{R}$, where $\mathbf{w}$ denotes its weights and $\mathbf{w}_l$ denotes the weights in the l-th layer. We assume that $f(\mathbf{w}, \cdot)$ is η-smooth and $\|\mathbf{w}_l\|_2$ is W_l-bounded.

5.1 GAN setting

- **Distribution**: $\mathcal{X} \subseteq [0,1]^d$ and $\mathcal{Y} = \{-1, 1\}$.
- **Deep neural classifier**: L-layer MLP or CNN $f(\mathbf{w}, \cdot) : \mathcal{Z} \to \mathbb{R}$, where $\mathbf{w}$ denotes its weights and $\mathbf{w}_l$ denotes the weights in the l-th layer. We assume that $f(\mathbf{w}, \cdot)$ is η-smooth and $\|\mathbf{w}_l\|_2$ is W_l-bounded.
- **Learning algorithm for the classifier**: the loss function is the cross-entropy loss and it is optimized by SGD. For the t-th step, we set the step size as $\frac{c}{\eta t}$. Besides, the total iteration number $T = O(m_T)$.

5.1 GAN setting

- **Distribution**: $\mathcal{X} \subseteq [0,1]^d$ and $\mathcal{Y} = \{-1, 1\}$.
- **Deep neural classifier**: L-layer MLP or CNN $f(\mathbf{w}, \cdot) : \mathcal{Z} \to \mathbb{R}$, where $\mathbf{w}$ denotes its weights and $\mathbf{w}_l$ denotes the weights in the l-th layer. We assume that $f(\mathbf{w}, \cdot)$ is η-smooth and $\|\mathbf{w}_l\|_2$ is W_l-bounded.
- **Learning algorithm for the classifier**: the loss function is the cross-entropy loss and it is optimized by SGD. For the t-th step, we set the step size as $\frac{c}{\eta t}$. Besides, the total iteration number $T = O(m_T)$.
- **Deep generative model**: GAN is parameterized by MLP and its architecture is the same as that in Theorem 19 of [6] (somewhat strong assumptions). In addition, we assume that each category is learned by a GAN, respectively.

Theorem 8 (Generalization bound for GAN)

Given a set S with m_S i.i.d. samples from any distribution $\mathcal{D}$ and an augmented set S_G with m_G i.i.d. examples sampled from the distribution $\mathcal{D}_G(S)$ learned by GANs, then for any fixed $\delta \in (0,1)$, with probability at least $1-\delta$, it holds that

$$\mathbb{E}|\textit{Gen-error}| \lesssim \begin{cases} \frac{1}{\sqrt{m_S}} & \text{if fix } W, L, d, \text{ let } m_G = 0, \\ \left(\frac{\log(m_S)}{m_S}\right)^{\frac{1}{4}} & \text{if fix } W, L, d, \text{ let } m_G = m^*_{G,\text{order}}, \\ dL^2 \left(\prod_{l=1}^{L} \|W_l\|_2\right)^2 & \text{if fix } m_S. \end{cases}$$

5.2 Theoretical result

Slow learning rate with GDA

When we perform GDA, the order with regard to m_S strictly becomes worse. Therefore, it implies that when m_S is rich, it is hopeless to boost the performance obviously by augmenting the train set based on GANs. On the contrary, GDA may make the generalization worse.

5.2 Theoretical result

Slow learning rate with GDA

When we perform GDA, the order with regard to m_S strictly becomes worse. Therefore, it implies that when m_S is rich, it is hopeless to boost the performance obviously by augmenting the train set based on GANs. On the contrary, GDA may make the generalization worse.

GDA matters a lot when overfitting happens

As the data dimension and model capacity become larger, the deep neural classifier gains terrible generalization performance. In this case, a constant-level improvement of generalization caused by GDA will be significant.

5.3 Experimental design

GANs are chosen to validate Theorem 8 empirically and the EDM is chosen to explore the ability of the diffusion model.

5.3 Experimental design

GANs are chosen to validate Theorem 8 empirically and the EDM is chosen to explore the ability of the diffusion model.

- We choose a "good" GAN (StyleGAN2-ADA) to verify that GANs can not improve the test performance obviously when the m_S is approximately large (with standard augmentation).

5.3 Experimental design

GANs are chosen to validate Theorem 8 empirically and the EDM is chosen to explore the ability of the diffusion model.

- We choose a "good" GAN (StyleGAN2-ADA) to verify that GANs can not improve the test performance obviously when the m_S is approximately large (with standard augmentation).
- We choose a "bad" GAN (DCGAN) to empirically verify that GANs can improve the test performance when m_S is small and awful overfitting happens (without standard augmentation).

5.3 Experimental design

GANs are chosen to validate Theorem 8 empirically and the EDM is chosen to explore the ability of the diffusion model.

- We choose a "good" GAN (StyleGAN2-ADA) to verify that GANs can not improve the test performance obviously when the m_S is approximately large (with standard augmentation).
- We choose a "bad" GAN (DCGAN) to empirically verify that GANs can improve the test performance when m_S is small and awful overfitting happens (without standard augmentation).
- We conduct experiments on the SOTA diffusion model (EDM) and suggest that diffusion models have a better $\mathcal{D}_{\mathrm{TV}}(\mathcal{D}, \mathcal{D}_G(S))$ than GANs.

5.4 Empirical results

Generator	Classifier	S.A.	GDA (m_G)					
			0	100k	300k	500k	700k	1M
cDCGAN [55]	ResNet18	×	85.76	86.8	87.83	87.59	87.52	86.47
		√	94.4	93.92	93.41	93.81	93.01	92.6
	ResNet34	×	85	86.9	87.93	87.56	87.17	86.28
		√	94.59	94.83	94.21	93.64	93.69	93.18
	ResNet50	×	82.85	87.49	88.59	86.67	86.3	85.2
		√	94.69	94.43	93.86	93.74	93.12	92.63
StyleGAN2-ADA [56]	ResNet18	×	85.76	90.22	91.33	91.37	91.25	91.38
		√	94.4	94.68	94.46	94.4	94.11	94.12
	ResNet34	×	85	90.24	91.23	91.45	91.56	90.91
		√	94.59	95.05	94.9	94.4	94.43	94.21
	ResNet50	×	82.85	90.85	92.29	92.29	92.29	91.61
		√	94.69	94.74	95.04	94.56	94.76	94.28
EDM [30]	ResNet18	×	85.76	92.8	94.87	95.43	96.24	96.28
		√	94.4	96.15	96.74	97.09	97.28	97.5
	ResNet34	×	85	93.42	94.93	95.59	96.14	96.44
		√	94.59	96.47	96.96	97.36	97.53	97.51
	ResNet50	×	82.85	93.29	95.29	95.95	96.1	96.64
		√	94.69	96.09	96.87	97.28	97.6	97.74

Figure: Accuracy on the CIFAR-10 test set.

Table of Contents

Conclusion

Our contributions

- We establish a general theoretical framework for the GDA in supervised learning.
- We particularize the general results to the binary Gaussian mixture model (bGMM) and generative adversarial nets (GANs).
- We conduct experiments to validate our theoretical findings.

Future works (from easy to hard)

- Other settings: semi-supervised learning, adversarial training, etc.
- General non-i.i.d. learning:
  - Dependence graph: improving Theorem 4 by sharper moment bounds.
  - Mixing process: non-stationary stability bounds.
- Understanding generative models: memorization, generalization, stability, etc.

References I

[1] Shekoofeh Azizi et al. "Synthetic Data from Diffusion Models Improves ImageNet Classification". In: *CoRR* abs/2304.08466 (2023).

[2] Zebin You et al. "Diffusion Models and Semi-Supervised Learners Benefit Mutually with Few Labels". In: *CoRR* abs/2302.10586 (2023).

[3] Zekai Wang et al. "Better Diffusion Models Further Improve Adversarial Training". In: *CoRR* abs/2302.04638 (2023).

[4] Olivier Bousquet, Yegor Klochkov, and Nikita Zhivotovskiy. "Sharper Bounds for Uniformly Stable Algorithms". In: *COLT*. Vol. 125. 2020, pp. 610–626.

[5] Rui-Ray Zhang and Massih-Reza Amini. "Generalization bounds for learning under graph-dependence: A survey". In: *CoRR* abs/2203.13534 (2022).

[6] Tengyuan Liang. "How Well Generative Adversarial Networks Learn Distributions". In: *Journal of Machine Learning Research* 22 (2021), 228:1–228:41.