Statistical guarantees for generative models

Meating on Math. Stat. 2023, CIRM, Luminy (France)

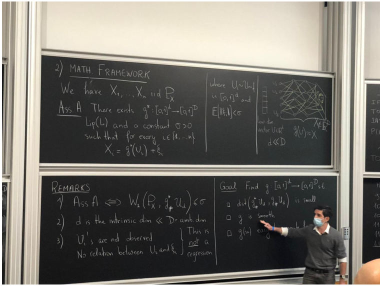
joint with

E. Vardanyan, S. Hunanyan, A. Minasyan, T. Galstyan

CREST, ENSAE, Institut Polytechnique de Paris

December 19, 2023

First episode (December 16, 2021)



2) MATH. FRAMEWORK

We have $X_1, \dots, X_n$ iid P_X

Ass. A There exists $g^*: [0,1]^d \to [0,1]^D$ Lip(L) and a constant $\sigma > 0$ such that for every $i \in \{1, \dots n\}$

$$X_i = g^*(U_i) + \xi_i$$

where $U_i \sim \mathcal{U}nif$ in $[0,1]^d$ and $\mathbb{E}[\|\xi_i\|] < \sigma$

REMARKS

1) Ass. A $\iff W_1(P_X, g^*_\# \mathcal{U}_d) \leq \sigma$

2) d is the intrinsic dim $\ll$ D= amb. dim

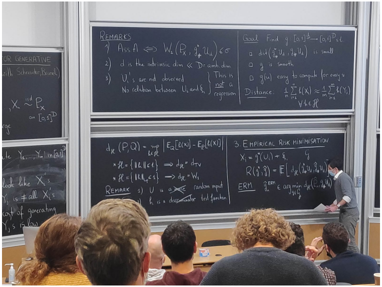
3) U_i's are not observed
No relation between U_i and ξ_i } This is not a regression

Goal Find $g: [0,1]^d \to [0,1]^D$ s.t.

□ $\mathrm{dist}(g^*_\# \mathcal{U}_d, g_\# \mathcal{U}_d)$ is small

□ g is smooth

□ g(u) eas



REMARKS

1) Ass. A $\iff W_1(P_X, g^*_\# \mathcal{U}_d) \leq \sigma$

2) d is the intrinsic dim $\ll$ D= amb. dim

3) U_i's are not observed
No relation between U_i and ξ_i } This is not a regression

Goal Find $g: [0,1]^d \to [0,1]^D$ s.t.

□ $\mathrm{dist}(g^*_\# \mathcal{U}_d, g_\# \mathcal{U}_d)$ is small

□ g is smooth

□ g(u) easy to compute for every u

Distance: $\frac{1}{n}\sum_{i=1}^{n} h(X_i) \approx \frac{1}{m}\sum_{j=1}^{m} h(Y_j)$ $\forall h \in \mathcal{H}$

$$d_{\mathcal{H}}(P,Q) = \sup_{h \in \mathcal{H}} \left| E_Q[h(X)] - E_P[h(X)] \right|$$

$* \ \mathcal{H} = \{\|h\|_\infty \leq 1\} \implies d_{\mathcal{H}} = d_{TV}$

$* \ \mathcal{H} = \{\|h\|_{Lip} \leq 1\} \implies d_{\mathcal{H}} = W_1$

REMARK 1) U is a ~~noise~~ random input

2) h is a ~~discriminator~~ test function

3. EMPIRICAL RISK MINIMISATION

$X_i = g^*(U_i) + \xi_i$ $\quad \mathcal{G}$

$R(g^*, \hat{g}) = \mathbb{E}\left[d_{\mathcal{H}}(g^*_\# \mathcal{U}_d, \hat{g}_\# \mathcal{U}_d) \right]$

ERM $\hat{g}_n^{ERM} \in \arg\min_{g \in \mathcal{G}} d_{\mathcal{H}}(\hat{P}_n, g_\# \mathcal{U}_d)$

GAN: generative adversarial networks

Goal: generate artificial images that look like true images.

The StyleGAN algorithm synthesizes photorealistic faces such as the examples above. Figure is from Karras et al. (2018).

GAN: generative adversarial networks

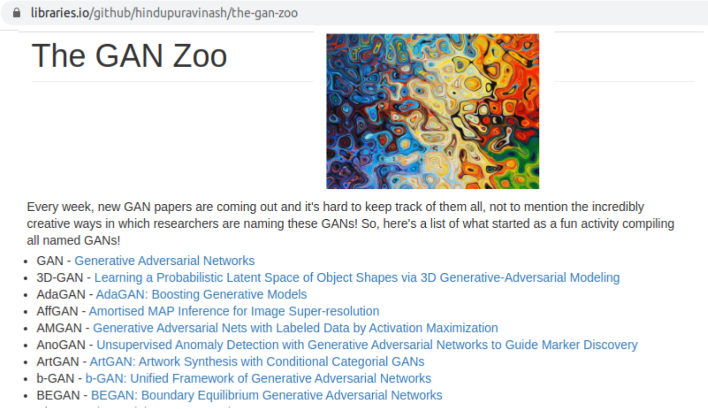
During the training process, the GAN needs a set of real faces to learn from.

Real faces such as these examples were used to train the algorithm. Figure is from the FFHQ dataset.

Since the seminal paper

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, Sh.; Courville, A.; Bengio, Y. (2014). Generative Adversarial Networks (PDF). NIPS 2014

many versions have been proposed



libraries.io/github/hindupuravinash/the-gan-zoo

The GAN Zoo

Every week, new GAN papers are coming out and it's hard to keep track of them all, not to mention the incredibly creative ways in which researchers are naming these GANs! So, here's a list of what started as a fun activity compiling all named GANs!

- GAN - Generative Adversarial Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling
- AdaGAN - AdaGAN: Boosting Generative Models
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AMGAN - Generative Adversarial Nets with Labeled Data by Activation Maximization
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorial GANs
- b-GAN - b-GAN: Unified Framework of Generative Adversarial Networks
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks

Questions for mathematicians

- Mathematical formalisation of the problem.
- Theoretical guarantees for widely used algorithms.
- Theoretical limits.
- Comparison and optimality of the algorithms.

Relevant prior work

- “Some theoretical properties of GANs”, Biau, Cadre, Sangnier, Tanielian, *Ann. Statist.* 48 (3), 2020.
- “Some Theoretical Insights into Wasserstein GANs”, Biau, Sangnier, Tanielian, *JMLR* 22, 2021.
- “How Well Generative Adversarial Networks Learn Distributions”, Tengyuan Liang, JMLR, 2021.
- “Statistical guarantees for generative models without domination”, N. Schreuder, V.-E. Brunel, A. D., ALT, 2021.
- “Rates of convergence for density estimation with GAN”, D. Belomestny, E. Moulines, A. Naumov, N. Puchkin, S. Samsonov, arXiv:2102.00199, 2021.
- “Generative Modeling with Denoising Auto-Encoders and Langevin Sampling”, A. Block, Y. Mroueh, A. Rakhlin arXiv:2002.00107, 2020.
- “Statistical Efficiency of Score Matching: The View from Isoperimetry”, F. Koehler, A. Heckett, A. Risteski, ICLR 2023.
- “The Speed of Mean Glivenko-Cantelli Convergence”, R. M. Dudley, Ann. Math. Statist. 40(1): 40-50 (February, 1969).

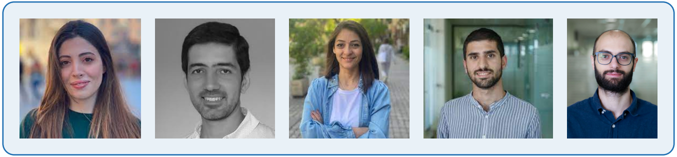
Problem formulation

Generative models are used for accomplishing the following task.

- Nature draws n independent vectors $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n$ from a distribution P^* defined on $\mathbb{R}^D$.
- We are given a noisy and contaminated version $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ of this sample.
- The goal is to design an algorithm that generates random vectors from a distribution $P_{\textsf{learner}}$ which is as close as possible to P^*.

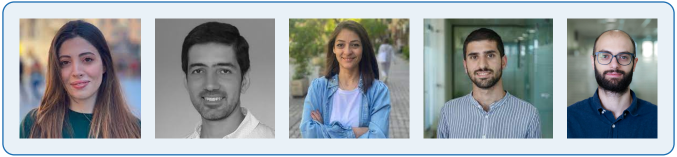
Poor generator 1

Training sample



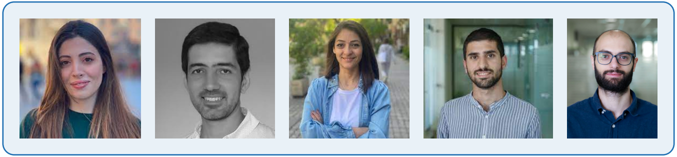
Poor generator 1

Training sample

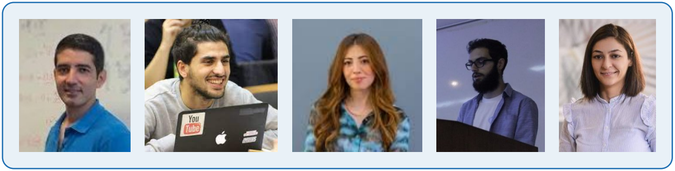


Poor generator 1

Training sample



Generated examples



Poor generator 1

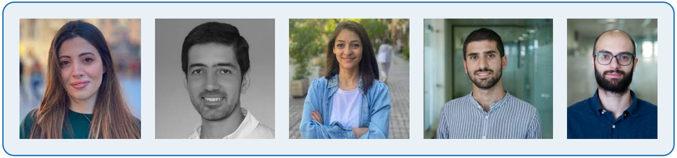
Training sample

Generated examples

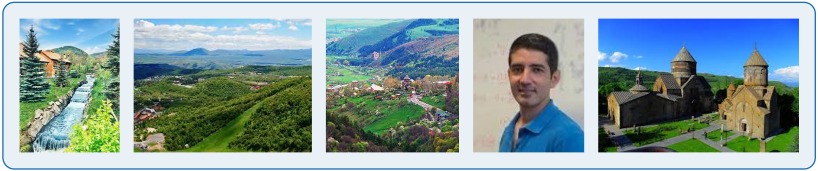
New example generation should be fast

Poor generator 2

Training sample

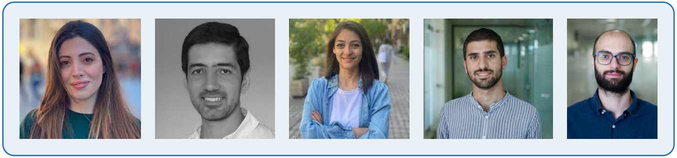


Generated examples



Poor generator 2

Training sample

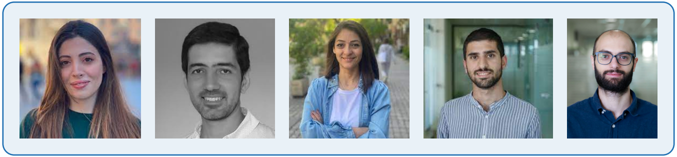


Generated examples

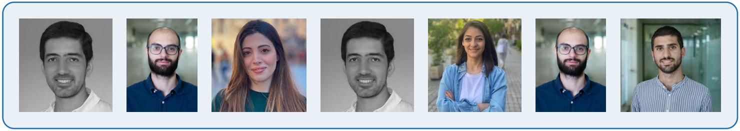
Distribution of the generated examples should be close to the data generating distribution.

Poor generator 3

Training sample

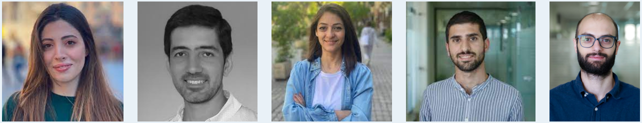


Generated examples

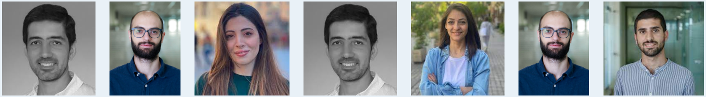


Poor generator 3

Training sample



Generated examples



Generator should avoid replication.

Desirable properties

This can be viewed as a distribution estimation problem with two requirements:

[R1] It should be easy to sample from P_{learner}.

[R2] P_{learner} should be "close" to P^* (the error has to admit an interpretation as a sampling error).

Desirable properties

This can be viewed as a distribution estimation problem with two requirements:

[R1] It should be easy to sample from P_{learner}.

[R2] P_{learner} should be "close" to P^* (the error has to admit an interpretation as a sampling error).

This formulation is incomplete since it allows to take the uniform distribution $\widehat{P}_n$ over the observations as $P_{\textsf{learner}}$ ($\widehat{P}_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{\boldsymbol{X}_i}$).

From a generative model perspective, $\widehat{P}_n$ is pointless: it does not yield examples that are different from the observed ones.

Desirable properties

This can be viewed as a distribution estimation problem with two requirements:

[R1] It should be easy to sample from P_{learner}.

[R2] P_{learner} should be "close" to P^* (the error has to admit an interpretation as a sampling error).

This formulation is incomplete since it allows to take the uniform distribution $\widehat{P}_n$ over the observations as $P_{\textsf{learner}}$ ($\widehat{P}_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{\boldsymbol{X}_i}$).

From a generative model perspective, $\widehat{P}_n$ is pointless: it does not yield examples that are different from the observed ones.

[R3] *Examples drawn from $P_{\textsf{learner}}$ should be different from those revealed to the learner.*

Notation

- $\mathcal{U}_d$ is the uniform distribution on the hyper-cube $[0,1]^d$.
- For convex $\mathcal{X} \subset \mathbb{R}^d$, $\mathsf{Lip}_L(\mathcal{X})$ is the set of all Lipschitz functions with a Lipschitz constant $\leq L$.
- For a function $g : \mathcal{X} \to \mathbb{R}$, $\|g\|_\infty = \max_{x \in \mathcal{X}} |g(x)|$.

Notation

- $\mathcal{U}_d$ is the uniform distribution on the hyper-cube $[0,1]^d$.
- For convex $\mathcal{X} \subset \mathbb{R}^d$, $\mathsf{Lip}_L(\mathcal{X})$ is the set of all Lipschitz functions with a Lipschitz constant $\leq L$.
- For a function $g : \mathcal{X} \to \mathbb{R}$, $\|g\|_\infty = \max_{x\in\mathcal{X}} |g(x)|$.
- For a distribution P defined on $(E, \mathcal{E})$ and a measurable $g : (E, \mathcal{E}) \mapsto (F, \mathcal{F})$, we denote by $g\sharp P$ the "push-forward" measure defined by

$$(g\sharp P)(A) = P\big(g^{-1}(A)\big), \qquad \forall A \in \mathcal{F}.$$

Manifold Assumption

In most applications, the ambient dimension D is very large.

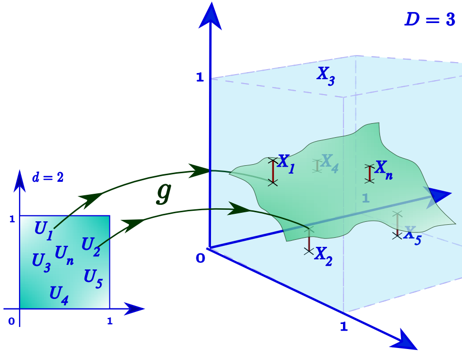
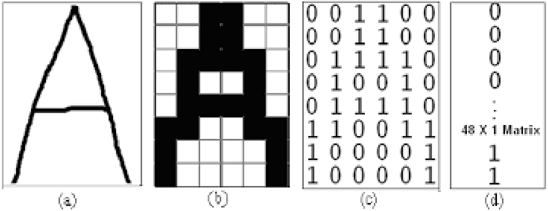


Figure: Illustration of the manifold assumption (**Ass. A**). Most $\boldsymbol{X}_i$'s are close to the manifold defined as the image of $[0, 1]^d$ by the map g.

Manifold Assumption

Ambient dimension versus latent dimension

In most applications, the ambient dimension D is very large.



$$\begin{matrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{matrix}$$

$$\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{matrix}$$

Figure: Vector representation of a black and white image: the ambient dimension is $D = 48$. Latent dimension is $d = 12$.

Manifold Assumption

Formal and extended version

Let $\sigma \geq 0$ be some constant.

Assumption A: There exists $g^* : [0,1]^d \to [0,1]^D$ (with $d \ll D$), random vectors $\boldsymbol{U}_1, \dots, \boldsymbol{U}_n \in \mathbb{R}^d$ and $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_n \in \mathbb{R}^D$ such that

- We have $\boxed{\boldsymbol{X}_i = g^*(\boldsymbol{U}_i) + \boldsymbol{\xi}_i}$ for every $i \in \{1, \dots, n\}$.
- $\boldsymbol{U}_i$ are iid uniformly distributed in $[0,1]^d$ ($\boldsymbol{U}_i \overset{\text{iid}}{\sim} \mathcal{U}_d$),
- $\max_{i=1,\dots,n} \mathbf{E}[\|\boldsymbol{\xi}_i\|_2] \leq \sigma$ for some $\sigma < \infty$.

The parameter σ, referred to as the noise magnitude, is unknown but assumed to be small.

Manifold Assumption

Formal and extended version

Let $\sigma \geq 0$ be some constant.

Assumption A: There exists $g^* : [0,1]^d \to [0,1]^D$ (with $d \ll D$), random vectors $\boldsymbol{U}_1, \dots, \boldsymbol{U}_n \in \mathbb{R}^d$ and $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_n \in \mathbb{R}^D$ such that

- We have $\boxed{\boldsymbol{X}_i = g^*(\boldsymbol{U}_i) + \boldsymbol{\xi}_i}$ for every $i \in \{1, \dots, n\}$.
- $\boldsymbol{U}_i$ are iid uniformly distributed in $[0,1]^d$ ($\boldsymbol{U}_i \overset{\text{iid}}{\sim} \mathcal{U}_d$),
- $\max_{i=1,\dots,n} \mathbf{E}[\|\boldsymbol{\xi}_i\|_2] \leq \sigma$ for some $\sigma < \infty$.

The parameter σ, referred to as the noise magnitude, is unknown but assumed to be small.

Alternatively, one can see this assumption merely as the definition of σ:

$$\sigma = \mathsf{W}_1(P_X, g^* \sharp \mathcal{U}_d)$$

Risk of a generator and ERM

- Let $\mathcal{G}$ be a set of smooth (at least Lipschitz) functions.
- For every candidate generator g—a measurable mapping from $[0,1]^d$ to $\mathbb{R}^D$—we define the risk

$$R_{P^*}(g) = \mathsf{W}_1\big(g\sharp\mathcal{U}_d, P^*\big). \tag{1}$$

- Our goal is to find a mapping

$$\widehat{G} : (\mathbb{R}^D)^n \to \mathcal{G} \quad (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n) \mapsto \widehat{g}_n, \tag{2}$$

such that $R_{P^*}(\widehat{g}_n)$ is as small as possible.

- We define the ERM, by

$$\widehat{g}^{\,\mathsf{ERM}}_{n,\mathcal{G}} \in \arg\min_{g\in\mathcal{G}} \mathsf{W}_1\big(g\sharp\mathcal{U}_d, \widehat{P}_n\big). \tag{ERM}$$

Main result 1

Theorem (P_{learner} is close to P^*)

Let Assumption A hold with $d > 2$ and let the coordinates g_j^* of g^* belong to $\mathrm{Lip}_L([0,1]^d)$ for some $L \geq 1$. Then, the ERM satisfies

$$\mathbf{E}[R_{P^*}(\widehat{g}_{n,\mathcal{G}}^{\text{ERM}})] \leq \underbrace{\inf_{g\in\mathcal{G}} R_{P^*}(g) + 2\sigma}_{\text{approx. error}} + \underbrace{\frac{c_{d,D}L}{n^{1/d}}}_{\text{generalisation error}} . \quad (3)$$

where c is a constant which depends only on d, D.

Left-inverse constrained ERM

An illustration

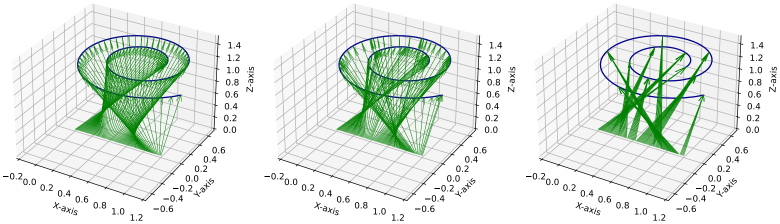


Figure: Generating points on a 2D spiral using a 1D latent space. The mapping $g : [0, 1] \to \mathbb{R}^2$ is represented by the green arrows. Each arrow indicates how points from the latent space are mapped to corresponding positions in the 2D spiral.

Left-inverse constrained ERM

Definition

We define the LIC-ERM, as a solution $\widehat{g}_{n,\mathcal{G}}^{\mathsf{LICERM}}$ to

$$\begin{aligned}
&\mathsf{minimise} \mathsf{W}_1\big(g\sharp\,\mathcal{U}_d, \widehat{P}_n\big) \qquad \text{(LICERM)}\\
&\text{subject to}\\
&\qquad g \in \mathcal{G} \text{ and}\\
&\qquad \exists\, h \in \mathsf{Lip}_{L_h}(\mathbb{R}^D) \text{ s.t. } h \circ g = \mathsf{Id}_d.
\end{aligned}$$

Main result 2

Theorem (P_{learner} is close to P^* and bounded away from $\widehat{P}_n$)

Let Assumption A hold with $d > 2$ and let the coordinates g_j^* of g^* belong to Lip$_L([0,1]^d)$ for some $L \geq 1$. Then, the LICERM satisfies

$$\mathbf{E}[R_{P^*}(\widehat{g}_{n,\mathcal{G}})] \leqslant \inf_{g \in \mathcal{G}_{L_h}} R_{P^*}(g) + 2\sigma + \frac{c_{d,D} L}{n^{1/d}},$$

$$\mathsf{W}_1(\widehat{g}_{n,\mathcal{G}} \sharp \mathcal{U}_d, \widehat{P}_n) \geqslant \frac{1}{2L_h(1 + c'_d n^{1/d})}.$$

The second inequality means that LIC-ERM is bounded away from the empirical distribution of the training sample. Thus, it avoids replication!

Relation to prior work

- [Biau et al., AoS 2020] establish large sample properties of the estimated distribution assuming that all the densities are dominated by a fixed known measure on a Borel subset of $\mathbb{R}^D$.
- When the admissible discriminators are neural networks with a given architecture, [Biau et al., arXiv 2020] obtains the parametric rate $n^{-1/2}$.
- [Luise et al., arXiv 2020] measure the quality of sampling through the Sinkhorn divergence (while we consider IPMs) and consider smoothness larger than $d/2$. The latter leads to parametric rates of convergence $n^{-1/2}$.

Conclusion

- We considered a general and nonparametric framework for learning generative models.
- Given high-dimensional data, we learn their distribution in order to sample new data points that resemble the training ones, while not being identical to those.
- A key point in our work is to leverage the fact that the distribution of the training samples, up to some approximation error and adversarial contamination, is supported by a low-dimensional smooth manifold.
- This allows us to alleviate the curse of dimensionality.
- We derived nonasymptotic bounds for the risk of our empirical risk minimizer, with rates of convergence that depend on the ambient dimension only through multiplicative constants.