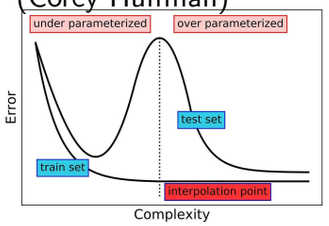
APPM 4490/5490
"Theory of Machine Learning"
Prof. Becker, spring 2024
Student projects

Commentary on Double Descent
(Corey Huffman)



Connections Between Differentially Private Learning and Online Learning
(Elias Lindgren)

Definition (Private Learning Algorithm) *A randomized learning algorithm*

$$A : (X \times \{\pm 1\})^m \to \{\pm 1\}^X$$

is ϵ, δ-differentially private if for every two samples $S, S' \in (X \times \{\pm 1\})^m$ that disagree on a single example, the output distributions A(S) and A(S') are (ϵ, δ)-indistingushable.

Littlestone Dimension of $\mathcal{H}$ is largest d s.t. $\mathcal{H}$ shatters all depth d trees

Finite Littlestone Dimension $\equiv$ online learnable

Understanding a Randomized Extension of Littlestone Dimension
(Mary Monroe)

Lemma 1. *For any hypothesis class $\mathcal{H}$, $M^*(\mathcal{H}) \geq RL(\mathcal{H})$.*

Proof. We construct an adversary who forces all randomized algorithms to make at least $RL(\mathcal{H})$ mistakes. Pick any finite tree T shattered by $\mathcal{H}$. Then the adversary constructs a random walk down

Continuum-armed Bandits: Incentivization and Non-Stationarity
(Amit Kiran Rege)

Theorem 1. *The expected regret for Algorithm (1) on the metric space (A, Φ), with ψ-covering A_0 of size $|A_0| \leqslant \lambda_d/\psi^d$ where d is the covering dimension of A, for some constant $\lambda_0 > 0$, is*

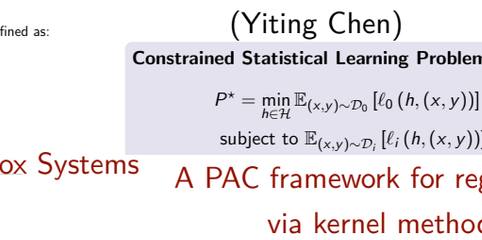
$$\mathbb{E}\left[R_T^A\right] \leqslant \lambda_e^r \cdot T^{(d+2)/(d+3)} \cdot L^{d/(d+3)} \cdot (\log T)^{1/(d+3)}$$

where $\lambda_e^r = \left(8\lambda_0^3\lambda_d\right)^{1/(d+3)}$. The expected compensation for the same setting, for some constant $\lambda > 0$ is

$$\mathbb{E}\left[C_T^A\right] \leqslant \lambda_e^c \cdot T^{(d+1)/(d+3)} \cdot L^{2d/(d+3)} \cdot (\log T)^{2/(d+3)}$$

where $\lambda_e^c = (4\sqrt{2})(8)^{(1-d)/(2d+6)}(\lambda_d)^{2/(d+3)}$

Explorations in SINDy
(Eric Flaska)



Active Learning
(K. Aditi)

Algorithm: **CAL**(n)
0. $m \leftarrow 0$, $t \leftarrow 0$, $V \leftarrow \mathbb{C}$
1. While $t < n$ and $m < 2^n$
2. $\quad m \leftarrow m + 1$
3. $\quad$ If $X_m \in \mathrm{DIS}(V)$
4. $\quad\quad$ Request label Y_m; let $V \leftarrow \{h \in V : h(X_m) = Y_m\}$, $t \leftarrow t + 1$
5. Return any $\hat{h} \in V$

B. Rademacher Processes and Data Dependent Bounds on Excess Risk

It is important to be able to upper bound the excess risk with bounds that are data-dependent and not distribution dependent since the joint distribution of $(X, Y), \mathcal{D}$ is unknown. The main idea of [3] is to characterize the complexity of class $\mathcal{F}$ by using Rademacher complexity. In general,

Learning Finite Automota
(Jasdeep Singh)

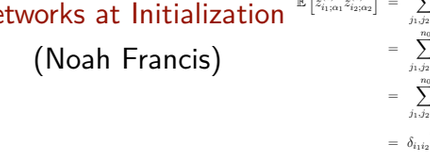
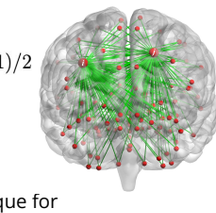


Figure 2: Nondeterministic Finite Automaton (NFA)

Network-Valued Regression
(Addie McCurdy)

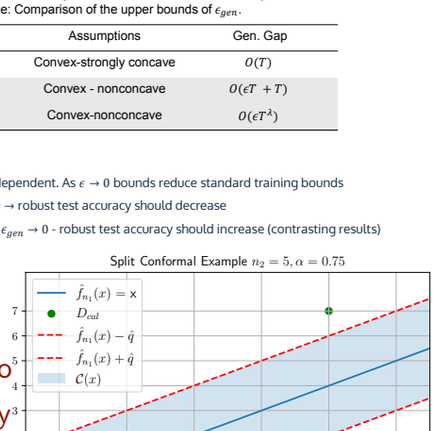


Proved:

- $\mathcal{L}_m$ *is a bounded, closed, and convex subset in* $\mathbb{R}^{m^2}$ *of dimension* $m(m-1)/2$
- $m_G(x) = P_{\mathcal{L}_m}(E[s_G(x)L])$
- $\hat{m}_G(x) = P_{\mathcal{L}_m}\left(\frac{1}{n}\sum_{k=1}^{n} s_{kG}(x)L_k\right)$

Thus, the population and sample global regressions exist and are unique for all x. (And it is a convex learning problem!)

Fractals in Gradient Descent Dynamics
(Joanna Parker)



Neural Networks as Gaussian Processes
(Walter Virany)

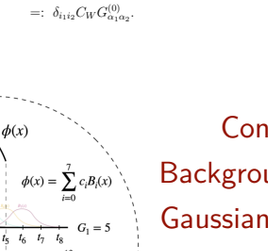


Figure: Ensemble of neural networks converges to a Gaussian process. Created using the `Neural Tangents` package ([Novak et al., 2019]).

Student backgrounds:

- Applied Math (BS/MS, MS, PhD)
- Math (BA)
- Computer Science (BS, PhD)
- Electrical Engineering (PhD)
- Aerospace Engineering (MS, PhD)
- Civil Engineering (PhD)
- Engineering Physics (BS)

Review of "Towards Size-Independent Generalization Bounds for Deep Operator Nets"
(Kal Parvanov)

Theorem (2.2)

Considering the same class of DeepONets as in Theorem 2.1 and using the Huber loss $\ell_{H,\delta}(x) := \begin{cases} \frac{1}{2}x^2 & \text{for } |x| \leq \delta \\ \delta \cdot (|x| - \frac{1}{2}\delta) & \text{for } |x| > \delta \end{cases}$ *with* $\delta = \left(\frac{1}{2}\right)^{n-1}$ *as the loss function. The expectation over data of the supremum of the generalization error over the above class of DeepONets can then be bounded by,* $\mathcal{O}\left(\frac{\mathcal{C}_{n,n-1}}{\sqrt{m}}\left(\prod_{j=2}^{n-1}\mathcal{C}_{-j,-j}\right)M_{X,B}M_{X,T}\right)$ *where $\mathcal{C}_{n,n-1}$ and $\mathcal{C}_{-j,-j}$ are the constants defined in Theorem 2.1.*

On the cross-validation bias due to unsupervised preprocessing
(Bart Chen)

Theorem 2. *Let each element of the matrix $X \in \mathbb{R}^{(n+m)\times p}$ and each coefficient β_j of $\boldsymbol{\beta} \in \mathbb{R}^p$ is drawn i.i.d. from a standard normal distribution, and let $Y = X\boldsymbol{\beta} \in \mathbb{R}^{n+m}$. If $\hat{j}, \hat{k} \in [p]$ are such that $\|X_{1:n+m,\hat{j}}\| \geq \|X_{1:n+m,k}\|$ for all $i \in [p]\backslash\{\hat{j}, \hat{k}\}$, and an OLS model with no intercept is trained on $\tilde{X} \in \mathbb{R}^{n\times 2}$ and $Y_{1:n}$ where $\tilde{X}_{1:n,1} = X_{1:n,\hat{j}} = \mathbf{p}_1$ and $\tilde{X}_{1:n,2} = X_{1:n,\hat{k}} = \mathbf{p}_2$, then the bias of the validation error with respect to the generalization error of the model is:*

$$bias = \mathbb{E}\left[(p-2)\left((X_{n+1,\hat{j}}^2 - 1)(Z_{j_0}^{(1)})^2 + (X_{n+1,\hat{k}}^2 - 1)(Z_{j_0}^{(2)})^2\right) - X_{n+1,\hat{j}}^2 - X_{n+1,\hat{k}}^2 + 2\right]$$

Where $j_0 \in [p]\backslash\{\hat{j}, \hat{k}\}$ is arbitrary, $Z_j^{(1)} = \frac{(\|\mathbf{p}_2\|^2\mathbf{p}_1^T - \mathbf{p}_1^T\mathbf{p}_2\mathbf{p}_2^T)\mathbf{p}_j}{\det(X^TX)}$, and $Z_j^{(2)} = \frac{(\|\mathbf{p}_1\|^2\mathbf{p}_2^T - \mathbf{p}_2^T\mathbf{p}_1\mathbf{p}_1^T)\mathbf{p}_j}{\det(X^TX)}$ for all $j \in [p]\backslash\{\hat{j}, \hat{k}\}$

Differentially Private and Consistent Multiclass Algorithms for Complex Objectives
(Drona Khurana)

Algorithm 4 Differentially Private LMO: $\tilde{\Omega}$

1: **Input:** Loss matrix $\mathbf{L} \in \mathcal{S}^{\mathbf{L}}$, Private CPE $\tilde{\eta} : \mathcal{X} \to \Delta_n$, Validation Data $S_2 = \{(x_{N+1}, y_{N+1}), \ldots, (x_{2N}, y_{2N})\}$
2: **Construct classifier:** $\tilde{g}_L(x) = argmin_{j\in[n]} \sum_{i=1}^{n} \tilde{\eta}_i(x) L_{n(i-1)+j}$
3: **Estimate confusion matrix:** $\hat{C}[\tilde{g}_L] = CME(\tilde{\eta}_L, S_2)$
4: Set $\sigma_1 = \frac{(\Delta_C)^2}{2\mu_1}$
5: **Privatize:** $vec(\tilde{C}[\tilde{g}_L]) = vec(\hat{C}[\tilde{g}_L]) + Z$, where $Z \sim \mathcal{N}(0_d, \sigma_1^2 \cdot I_{d\times d})$
6: **Project:** $\tilde{\mathbf{\Gamma}}_L = proj_{\Delta_d}(\tilde{C}[\tilde{g}_L])$
7: **Output:** $\tilde{g}_L, \tilde{\mathbf{\Gamma}}_L$

Probably Approximately Correct Constrained Learning
(Yiting Chen)

Constrained Statistical Learning Problem

$$P^\star = \min_{h\in\mathcal{H}} \mathbb{E}_{(x,y)\sim\mathcal{D}_0}\left[\ell_0\left(h,(x,y)\right)\right]$$
$$\text{subject to } \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\left[\ell_i\left(h,(x,y)\right)\right] \leq c_i,\ i = 1, \cdots, k,$$

Effective Theory of Deep Linear Networks at Initialization
(Noah Francis)

$$\begin{aligned}\mathbb{E}\left[z_{i_1;\alpha_1}^{(1)} z_{i_2;\alpha_2}^{(1)}\right] &= \sum_{j_1,j_2=1}^{n_0} \mathbb{E}\left[W_{i_1j_1}^{(1)} x_{j_1;\alpha_1} W_{i_2j_2}^{(1)} x_{j_2;\alpha_2}\right] \\ &= \sum_{j_1,j_2=1}^{n_0} \mathbb{E}\left[W_{i_1j_1}^{(1)} W_{i_2j_2}^{(1)}\right] x_{j_1;\alpha_1} x_{j_2;\alpha_2} \\ &= \sum_{j_1,j_2=1}^{n_0} \delta_{i_1i_2}\delta_{j_1j_2}\frac{C_W}{n_0} x_{j_1;\alpha_1} x_{j_2;\alpha_2} \\ &= \delta_{i_1i_2}\frac{C_W}{n_0}\sum_{j=1}^{n_0} x_{j;\alpha_1} x_{j;\alpha_2} \\ &=: \delta_{i_1i_2} C_W G_{\alpha_1\alpha_2}^{(0)}.\end{aligned}$$

Stability based Generalization Bounds for Adversarial Training
(Karan Muvvala)

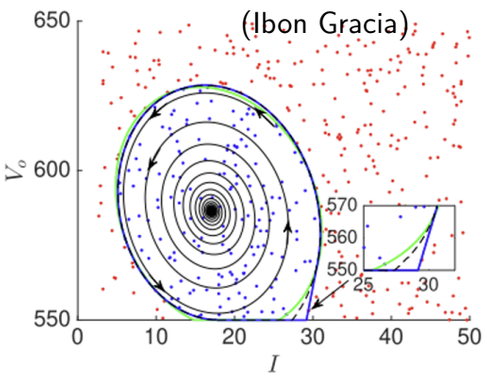
Table: Comparison of the upper bounds of ϵ_{gen}.

Paper	Assumptions	Gen. Gap
Farnia et al. [4]	Convex-strongly concave	$O(T)$
This work [1]	Convex - nonconcave	$O(\epsilon T + T)$
Alternate bound	Convex-nonconcave	$O(\epsilon T^4)$

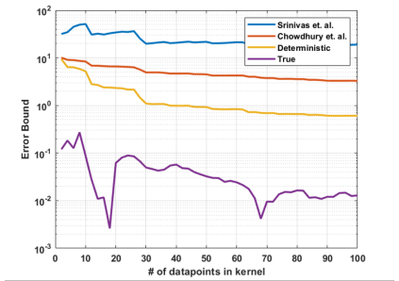
Takeaways:

- This project: Bounds are ϵ-dependent. As $\epsilon \to 0$ bounds reduce standard training bounds
- As $n \to \infty$ [1] suggests $O(\epsilon T) \to$ robust test accuracy should decrease
- Alternate bound: as $n \to \infty$, $\epsilon_{gen} \to 0$ - robust test accuracy should increase (contrasting results)

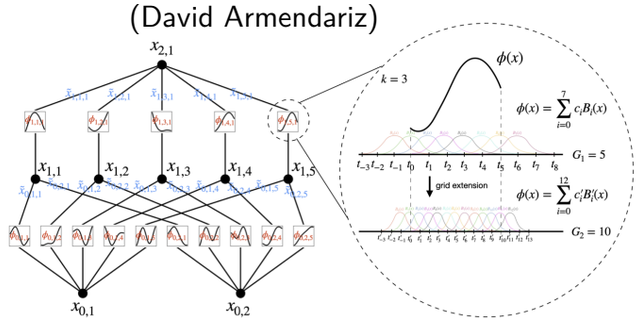
Data-driven Invariant Sets of Black-Box Systems
(Ibon Gracia)



A PAC framework for regression... via kernel methods
(Robert Reed)



Kolmogorov-Arnold Networks
(David Armendariz)



Conformal Prediction: Background and Comparison to Gaussian Processes Uncertainty Quantification
(Tyler Jensen)