Introduction

phenomenon: state-of-art models are over-parameterized.

data interpolation, memorization → shows in

contradiction: classical theory of generalization V.S. modern ML practice

extracting patterns
trade-offs between
less error on training sets and generalization abilities.

memorize everything needed.
performs good on training sets
with optimal generalization errors

explanation: the insufficient data hurdles the accuracy of model.

long-tailed datasets.

The aggregate of small subsets overweigh the larger ones.

"Fragmented" dataset.

For small subsets, its better to just remember them.

a little bit like "rules" and "exceptions" in life.

question: if memorizing these "exceptions" is a must to reach close-to-optimal generalization error?

① the algorithm can't decide the borderline between memorizing and not-memorizing

② the long-tailed dataset → even the smallest subsets can count when aggregating together.

∵ ①, ②

∴ Need to memorize

results.

$$\overline{err}(\pi, F, A) \geq opt(\pi, F) + \tau_1 \cdot E[err_{n_S}(A, 1)]$$

expected generalization error

min achievable error

num of examples appear only once and mislabeled.

mistakes on "memory"

$$\tau_1 = \frac{E_{a\sim\bar{\pi}^N}[2^2 \cdot (1-\alpha)^{n-1}]}{E_{a\sim\bar{\pi}^N}[2 \cdot (1-\alpha)^{n-1}]}$$

Memorization in Unstructed Classification

$[n] = \{1, \ldots, n\}$ $n \in \{1, 2, 3 \cdots\}$

$1(O)$: under the condition O, whether the event will happen or not can be expressed by '0' for happen and '1' for not happen.

dataset $S = (x_1, y_1) \cdots (x_n, y_n)$ X_S = all points in S.

Probablity destribution of $F(x)$:

value of $F(x)$ F is a function on X.

$D_x \sim D[F(x) \mid x \in O]$ under the condition o

$x \in X$ for all x subjective to distribution D.

for all the x in X which subjective to distribution D. and under condition o.

the distribution of $F(x)$, for F is a function on X.

$TV(D_1, D_2)$: variation distance between D_1, D_2

function $h: X \to Y$ prediction function

dataset $S: (x_1, y_1), \ldots, (x_n, y_n)$ x for points y for labels.

$err_p(h) := E_{(x,y)\sim P}[h(x) \neq y]$ the expectation of the prediction mistakes of given x.

For an algorithm A, the expected generalization error:

$err_p(A, S) := \underset{h\sim A(S)}{E}[err_p(h)]$ the expectation of $err_p(h)$

h being the output of the algorithm.

For an algorithm A on any sample S in P, the expected generalization error.

$err_p(A) := E_{S\sim P^n}[err_p(A, S)]$ the expectation of $err_p(A)$

$err_p(h)$ is the indicator of prediction action.

$\downarrow E$ E for all predict. action. for one $h(x)$

$err_p(A, S)$ is the indicator of the algorithm.

E for all the possible $h(x)$

$\downarrow E$

$err_p(A)$ is the indicator of all of the data points.

E for all the possible sample S on P.

Problem setup.

$|X| = N$ data domain $\quad$ $|Y| = m$ label domain.

Lemma 2.1.

$$E[D(x_i) \mid U = V] = \frac{E_{\alpha\sim\bar{\pi}^N}[\alpha^{l+1}\cdot(1-\alpha)^{n-l}]}{E_{\alpha\sim\bar{\pi}^N}[\alpha^{l}\cdot(1-\alpha)^{n-l}]}$$

$D \sim D_\pi^X$, $U \sim D^n$ Distribution over $D(x)$ $(x_1, x_2 \cdots x_n)$

(Each point in U) $\sim D$

l: in set V, x shows l times

$\bar{\pi}^N$ the marginal distribution

in condation $U = V$, the expectation of $D(x)$ can be represented as ...

generalization error on a randomly choosen problem:

$$\overline{err}(\pi, F, A) := \underset{D\sim D\ f\sim F}{E}[err_{D,f}(A)]$$

for the whole data domain and all the prediction functions.

dataset $S \in (X \times Y)^n$. $l \in [n]$

$X_{S=l}$: set of points shows l times in S.

$h: X \to Y$

$$err_S(h, l) := |\{x \in X_{S=l} \mid h(x) \neq y\}|$$

the number of x for which $h(x) \neq y$ and x belongs to $X_{S=l}$

$$err_S(A, l) := \underset{h\sim A(S)}{E}[err_S(h, l)]$$ for all of the prediction function h for Algorithm A.

For all the algorithms and datasets :

(D, f, S) : $D \sim D_\pi^X$ $f \sim F$ $S \sim (D, f)^n$ G is the distribution of (D, f, S)

D_π^X: the data domain; F: all the labeling functions; $(D, f)^n$: the datasets on domain D and using labeling function f

$S \in (X \times Y)^n$ — the f only define the labeling function; S is still a dataset.

For $Z \in (X \times Y)^n$ $G(|Z) := \underset{(D,f,S)\sim G}{D}[(D, f) \mid S = Z]$

the expected error of A condationed on dataset $= Z$

$$\overline{err}(\pi, F, A|Z) := \underset{(D,f)\sim G(Z)}{E}[err_{D,f}(A, Z)]$$

for all the datasets and labeling functions.

$$\min[\overline{err}(\pi, F, A|Z)] = \overline{err}(\pi, F, A'|Z) = opt(\pi, F|Z)$$

for all the algorithm As.

Theorem 2.3. $\overline{err}(\pi, F, A|z) \geq opt(\pi, F|z) + \sum_{l\in[n]} \tau_l \cdot errn_z(A, l)$

The not-fitting error of — min — $l=1,2,3\cdots n$ — not-fitting error For algorithm A and x showing l times

prior distribution π

labeling function $f \in F$

Algorithm A

dataset $z = (D, f)$

All of the mistakes $\geq$ min mistakes + number of all mistakes on points showing $1, 2, 3 \cdots n$ times.

$$\tau_l := \frac{E_{a\sim\bar{\pi}^N}[a^{l+1}\cdot(1-a)^{n-l}]}{E_{a\sim\bar{\pi}^N}[a^{l}\cdot(1-a)^{n-l}]}$$

The number of singleton is determined by "How long the tail is".

the fraction of examples in frequency range $[\beta_1, \beta_2]$:

$$Weight(\bar{\pi}^N, [a, \beta]) := \underset{D\sim D_\pi^x}{E}\left[\sum_{x\in X} D(x) \cdot \underbrace{1(D(x)\in[\beta_1, \beta_2])}_{\{0,1\}\text{ indicator}}\right]$$

$D(x)$: distribution

$\{0,1\}$ indicator: in range $\to 1$; not in range $\to 0$

Whether the data is counted depends on the data in the range or not.

$$= N \cdot \underset{a\sim\bar{\pi}^N}{E}[a \cdot 1(a\in[\beta_1, \beta_2])]$$

Get the data out of $D(x)$

divide the $D(x)$ into data (N) and distribution E.

The expected number of singleton points:

$$single(\bar{\pi}^N) := \underset{D\sim D, V\sim D^n}{E}[1_{X_V=1}]$$

the expectation for the number of all the "single" points

$$= \mathop{E}_{D\sim D_X^\pi}\left[\underbrace{\sum_{x\in X}\Pr_{V\sim D^n}[x\in X_{V=1}]}_{\text{the add results of all the prob of all of the single points}}\right]$$

$$= \mathop{E}_{D\sim D_X^\pi}\left[\sum_{x\in X} n\cdot \underbrace{D(x)}_{\text{distribution of } x}\underbrace{(1-D(x))^{n-1}}_{\text{distribution of "not } x\text{"}}\right]$$

$$= \sum_{x\in X} n \mathop{E}_{D\sim D_X^\pi}\left[D(x)(1-D(x))^{n-1}\right]$$

$$\sum_{x\in X} E[D(x)] = N\cdot \mathop{E}_{a\sim \partial x}[a]$$

$$= nN\cdot \mathop{E}_{a\sim \bar{\pi}^N}[a(1-a)^{n-1}]$$

for $a < \frac{1}{n}$, $(1-a)^{n-1} \geq \frac{1}{3}$

$$\text{single}(\bar{\pi}^N) \geq nN\cdot \mathop{E}_{a\sim\bar{\pi}^N}[\underbrace{a(1-a)^{n-1}}_{\geq\frac{1}{3}}\cdot \underbrace{1(a\in[0,\tfrac{1}{n}])}_{\text{add range for E}}]$$

$$\| $$

$$\frac{n}{3}N\mathop{E}_{a\sim\bar{\pi}^N}[a\cdot 1(a\in[0,\tfrac{1}{n}])]$$

$$\text{Weight}(\bar{\pi}^N, [\alpha,\beta]) = N\cdot \mathop{E}_{a\sim\bar{\pi}^N}[a\cdot 1(a\in[\beta_1,\beta_2])]$$

$$\therefore \frac{n}{3}N\mathop{E}_{a\sim\bar{\pi}^N}[a\cdot 1(a\in[0,\tfrac{1}{n}])] = \frac{n}{3}\,\text{weight}(\bar{\pi}^N,[0,\tfrac{1}{n}])$$

$$\text{single}(\bar{\pi}^N) \geq \frac{n}{3}\,\text{weight}(\bar{\pi}^N,[0,\tfrac{1}{n}])$$

the cost of not fitting any of the singleton examples is at least the weight of frequency of $\frac{1}{n}$

$$\text{Loostab}(P,A) := \frac{1}{n}\sum_{i\in[n]}\mathop{E}_{S\sim P^n}\left[\left|\underbrace{\Pr_{h\sim A(S)}[h(x_i)=y_i]}_{\text{the probablity of "correct" on dataset S. using algorithm A}} - \underbrace{\Pr_{h\sim A(S^{\setminus i})}[h(x_i)=y_i]}_{\text{the probablity of "correct" on dataset S. without point i using algorithm A}}\right|\right]$$

$i = 1, 2, 3, \cdots n.$

$$mem(A,S,i) := \max\left\{0, \Pr_{h\sim A(S)}[h(x_i)=y_i] - \Pr_{h\sim A(S^{\setminus i})}[h(x_i)=y_i]\right\}$$

the probablity of "correct" on dataset S. using algorithm A

the probablity of "correct" on dataset S. without point i using algorithm A

$$\min(mem(A,S,i)) = 0$$

mem evaluate the increase of prediction correctness

"How large can the correctness increase by 'knowing' point x_i."

r-memorization bounded: $x_i \in X_{S=1}$

for singletons

$$mem(A,S,i) \le r$$

"The maximum enhancement on correctness by memorizing points in $X_{S=1}$."

$$\frac{1}{n}\mathop{E}_{S\sim p^n}\left[\sum_{i\in[n]} mem(A,S,i)\right] = \frac{1}{n}\sum_{i\in[n]}\mathop{E}_{S\sim p^n}\left[\max\left\{0, \Pr_{h\sim A(S)}[h(x_i)=y_i] - \Pr_{h\sim A(S^{\setminus i})}[h(x_i)=y_i]\right\}\right]$$

$$Loostab(P,A) := \frac{1}{n}\sum_{i\in[n]}\mathop{E}_{S\sim p^n}\left[\left|\Pr_{h\sim A(S)}[h(x_i)=y_i] - \Pr_{h\sim A(S^{\setminus i})}[h(x_i)=y_i]\right|\right]$$

$$e = \Pr_{h\sim A(S)}[h(x_i)=y_i] - \Pr_{h\sim A(S^{\setminus i})}[h(x_i)=y_i]$$

$\max\{0,e\} \in [0,e]$

$|e| \in [-e,e]$

for $e \ge 0$ $\max\{0,e\} = |e|$

for $e < 0$ $\max\{0,e\} = 0$, $|e| = -e$, $0 < -e$

$$\Rightarrow \frac{1}{n}\mathop{E}_{S\sim p^n}\left[\sum_{i\in[n]} mem(A,S,i)\right] \le Loostab(P,A)$$

Lemma 4.2.

$$\underset{\substack{f\sim F.\\ S\sim(D.f)^n}}{E}[\mathrm{err}_S(A.l)] \geq \underset{\substack{f\sim F.\\ S\sim(D.f)^n}}{E}\left[\sum_{i\in[n],\, x_i\in X_{S=1}} \Pr_{h\sim A(S^{\setminus i})}[h(x_i)\neq y_i] - \mathrm{mem}(A.S.i)\right]$$

expectation of error Algorithm A made on points only showing l time.

The probablity of error for singletons on dataset S removing x_i for Algorithm A.

The correctness increasement for memorizing x_i

$$\mathrm{err}_S(h.l) := |\{x \in X_{S=1} \mid h(x)\neq y\}|$$

$$\mathrm{err}_S(A.l) := \underset{h\sim A(S)}{E}[\mathrm{err}_S(h.l)] =$$

$$\underset{\substack{f\sim F.\\ S\sim(D.f)^n}}{E}[\mathrm{err}_S(A.l)] = \underset{\substack{f\sim F\\ S\sim(D,f)^n}}{E}\left[\underset{h\sim A(S)}{E}[\,|\{x\in X_{S=1}\mid h(x)\neq y\}|\,]\right]$$

The number of error on singletons for every algorithm on S. every labeling function f and every dataset S.