Generalization in Deep Learning

Peter Bartlett Alexander Rakhlin

UC Berkeley

MIT

May 28, 2019

Outline

What is Generalization?

What is Generalization?

$$\log(1+2+3) = \log(1) + \log(2) + \log(3)$$

What is Generalization?

$$\log(1+2+3) = \log(1) + \log(2) + \log(3)$$

$$\log(1+1.5+5) = \log(1) + \log(1.5) + \log(5)$$

$$\log(2+2) = \log(2) + \log(2)$$

$$\log(1+1.25+9) = \log(1) + \log(1.25) + \log(9)$$

Outline

Statistical Learning Theory

Aim: Predict an outcome y from some set $\mathcal{Y}$ of possible outcomes, on the basis of some observation x from a feature space $\mathcal{X}$.

Use *data set* of n pairs

$$\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$

to choose a function $f : \mathcal{X} \to \mathcal{Y}$ so that, for subsequent (x, y) pairs, $f(x)$ is a good prediction of y.

Formulations of Prediction Problems

To define the notion of a 'good prediction,' we can define a **loss function**

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}.$$

$\ell(\hat{y}, y)$ is cost of predicting $\hat{y}$ when the outcome is y.

Aim: $\ell(f(x), y)$ small.

Formulations of Prediction Problems

Example

In *pattern classification* problems, the aim is to classify a pattern x into one of a finite number of classes (that is, the label space $\mathcal{Y}$ is finite). If all mistakes are equally bad, we could define

$$\ell(\hat{y}, y) = \mathbb{I}\{\hat{y} \neq y\} = \begin{cases} 1 & \text{if } \hat{y} \neq y, \\ 0 & \text{otherwise.} \end{cases}$$

Example

In a *regression* problem, with $\mathcal{Y} = \mathbb{R}$, we might choose the quadratic loss function, $\ell(\hat{y}, y) = (\hat{y} - y)^2$.

Probabilistic Assumptions

Assume:

- There is a probability distribution P on $\mathcal{X} \times \mathcal{Y}$,
- The pairs $(X_1, Y_1), \ldots, (X_n, Y_n), (X, Y)$ are chosen independently according to P

The aim is to choose f with small *risk*:

$$L(f) = \mathbb{E}\ell(f(X), Y).$$

For instance, in the pattern classification example, this is the misclassification probability.

$$L(f) = L_{01}(f) = \mathbb{E}\mathbf{I}\{f(X) \neq Y\} = \Pr(f(X) \neq Y).$$

Why not estimate the underlying distribution P (or $P_{Y|X}$) first?

This is in general a harder problem than prediction.

Key difficulty: our goals are in terms of unknown quantities related to unknown P. Have to use empirical data instead. Purview of statistics.

For instance, we can calculate the *empirical loss* of $f : \mathcal{X} \to \mathcal{Y}$

$$\widehat{L}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f(X_i))$$

The function $x \mapsto \widehat{f}_n(x) = \widehat{f}_n(x; X_1, Y_1, \ldots, X_n, Y_n)$ is random, since it depends on the random data $\mathcal{S} = (X_1, Y_1, \ldots, X_n, Y_n)$. Thus, the risk

$$\begin{aligned} L(\widehat{f}_n) &= \mathbb{E}\left[\ell(\widehat{f}_n(X), Y) | \mathcal{S}\right] \\ &= \mathbb{E}\left[\ell(\widehat{f}_n(X; X_1, Y_1, \ldots, X_n, Y_n), Y) | \mathcal{S}\right] \end{aligned}$$

is a random variable. We might aim for $\mathbb{E}L(\widehat{f}_n)$ small, or $L(\widehat{f}_n)$ small with high probability (over the training data).

Quiz: what is random here?

1. $\widehat{L}(f)$ for a given fixed f
2. $\widehat{f}_n$
3. $\widehat{L}(\widehat{f}_n)$
4. $L(\widehat{f}_n)$
5. $L(f)$ for a given fixed f

Theoretical analysis of performance is typically easier if $\widehat{f}_n$ has closed form (in terms of the training data).

E.g. ordinary least squares $\widehat{f}_n(x) = x^\top (X^\top X)^{-1} X^\top Y$.

Unfortunately, most ML and many statistical procedures are not explicitly defined but arise as

- solutions to an optimization objective (e.g. logistic regression)
- as an iterative procedure without an immediately obvious objective function (e.g. AdaBoost, Random Forests, etc)

The Gold Standard

Within the framework we set up, the smallest expected loss is achieved by the *Bayes optimal* function

$$f^* = \arg\min_f L(f)$$

where the minimization is over all (measurable) prediction rules $f : \mathcal{X} \to \mathcal{Y}$.

The value of the lowest expected loss is called the *Bayes error*:

$$L(f^*) = \inf_f L(f)$$

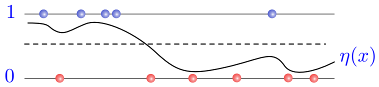
Of course, we cannot calculate any of these quantities since P is unknown.

Bayes Optimal Function

Bayes optimal function f^* takes on the following forms in these two particular cases:

- Binary classification ($\mathcal{Y} = \{0, 1\}$) with the indicator loss:

$$f^*(x) = \mathbb{I}\{\eta(x) \geq 1/2\}, \quad \text{where} \quad \eta(x) = \mathbb{E}[Y|X = x]$$



- Regression ($\mathcal{Y} = \mathbb{R}$) with squared loss:

$$f^*(x) = \eta(x), \quad \text{where} \quad \eta(x) = \mathbb{E}[Y|X = x]$$

The big question: is there a way to construct a learning algorithm with a guarantee that

$$L(\widehat{f_n}) - L(f^*)$$

is small for large enough sample size n?

Consistency

An algorithm that ensures

$$\lim_{n\to\infty} L(\widehat{f}_n) = L(f^*) \qquad \text{almost surely}$$

is called *universally consistent*. Consistency ensures that our algorithm is approaching the best possible prediction performance as the sample size increases.

The good news: consistency is possible to achieve.

- easy if $\mathcal{X}$ is a finite or countable set
- not too hard if $\mathcal{X}$ is infinite, and the underlying relationship between x and y is "continuous"

The bad news...

In general, we cannot prove anything quantitative about $L(\widehat{f}_n) - L(f^*)$, unless we make further assumptions (incorporate *prior knowledge*).

"No Free Lunch" Theorems: unless we posit assumptions,

- For any algorithm $\widehat{f}_n$, any n and any $\epsilon > 0$, there exists a distribution P such that $L(f^*) = 0$ and

$$\mathbb{E}L(\widehat{f}_n) \geq \frac{1}{2} - \epsilon$$

- For any algorithm $\widehat{f}_n$, and any sequence a_n that converges to 0, there exists a probability distribution P such that $L(f^*) = 0$ and for all n

$$\mathbb{E}L(\widehat{f}_n) \geq a_n$$

is this really “bad news”?

Not really. We always have some domain knowledge.

Two ways of incorporating prior knowledge:

- Direct way: assumptions on distribution P (e.g. margin, smoothness of regression function, etc)
- Indirect way: redefine the goal to perform as well as a reference set $\mathcal{F}$ of predictors:

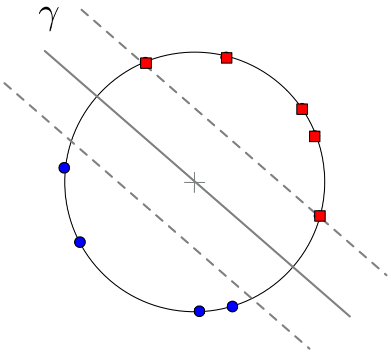
  $$L(\widehat{f}_n) - \inf_{f \in \mathcal{F}} L(f)$$

  $\mathcal{F}$ encapsulates our *inductive bias*.

We often make both of these assumptions.

Outline

Perceptron



Perceptron

$(x_1, y_1), \ldots, (x_T, y_T) \quad \in \mathcal{X} \times \{\pm 1\}$ (T may or may not be same as n)

Maintain a hypothesis $w_t \in \mathbb{R}^d$ (initialize $w_1 = \mathbf{0}$).

On round t,

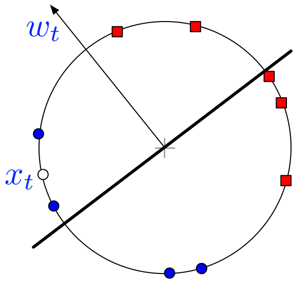
- Consider (x_t, y_t)
- Form prediction $\widehat{y}_t = \mathrm{sign}(\langle w_t, x_t \rangle)$
- If $\widehat{y}_t \neq y_t$, update

$$w_{t+1} = w_t + y_t x_t$$

  else

$$w_{t+1} = w_t$$

Perceptron



For simplicity, suppose all data are in a unit ball, $\|x_t\| \leq 1$.

Margin with respect to $(x_1, y_1), \ldots, (x_T, y_T)$:

$$\gamma = \max_{\|w\|=1} \min_{i \in [T]} \left(y_i \langle w, x_i \rangle\right)_+ ,$$

where $(a)_+ = \max\{0, a\}$.

Theorem (Novikoff '62)**.**

Perceptron makes at most $1/\gamma^2$ mistakes (and corrections) on **any** sequence of examples with margin γ.

Proof: Let m be the number of mistakes after T iterations. If a mistake is made on round t,

$$\|w_{t+1}\|^2 = \|w_t + y_t x_t\|^2 \leq \|w_t\|^2 + 2y_t \langle w_t, x_t \rangle + 1 \leq \|w_t\|^2 + 1.$$

Hence,

$$\|w_T\|^2 \leq m.$$

For optimal hyperplane w^*

$$\gamma \leq \langle w^*, y_t x_t \rangle = \langle w^*, w_{t+1} - w_t \rangle .$$

Hence (adding and canceling),

$$m\gamma \leq \langle w^*, w_T \rangle \leq \|w_T\| \leq \sqrt{m}.$$

Recap

For any T and $(x_1, y_1), \ldots, (x_T, y_T)$,

$$\sum_{t=1}^{T} \mathbf{I}\{y_t \langle w_t, x_t \rangle \leq 0\} \leq \frac{D^2}{\gamma^2}$$

where $\gamma = \gamma(x_{1:T}, y_{1:T})$ is margin and $D = D(x_{1:T}, y_{1:T}) = \max_t \|x_t\|$.

Let w^* denote the max margin hyperplane, $\|w^*\| = 1$.

Consequence for i.i.d. data (I)

Do *one pass* on i.i.d. sequence $(X_1, Y_1), \ldots, (X_n, Y_n)$ (i.e. $T = n$).

Claim: expected indicator loss of randomly picked function $x \mapsto \langle w_\tau, x \rangle$ ($\tau \sim \text{unif}(1, \ldots, n)$) is at most

$$\frac{1}{n} \times \mathbb{E}\left[\frac{D^2}{\gamma^2}\right].$$

Proof: Take expectation on both sides:

$$\mathbb{E}\left[\frac{1}{n}\sum_{t=1}^{n} \mathbf{I}\{Y_t \langle w_t, X_t\rangle \le 0\}\right] \le \mathbb{E}\left[\frac{D^2}{n\gamma^2}\right]$$

Left-hand side can be written as (recall notation $\mathcal{S} = (X_i, Y_i)_{i=1}^n$)

$$\mathbb{E}_\tau \mathbb{E}_\mathcal{S} \mathbf{I}\{Y_\tau \langle w_\tau, X_\tau\rangle \le 0\}$$

Since w_τ is a function of $X_{1:\tau-1}, Y_{1:\tau-1}$, above is

$$\mathbb{E}_\mathcal{S} \mathbb{E}_\tau \mathbb{E}_{X,Y} \mathbf{I}\{Y \langle w_\tau, X\rangle \le 0\} = \mathbb{E}_\mathcal{S}\mathbb{E}_\tau L_{01}(w_\tau)$$

Claim follows.

NB: To ensure $\mathbb{E}[D^2/\gamma^2]$ is not infinite, we assume margin γ in P.

Consequence for i.i.d. data (II)

Now, rather than doing one pass, cycle through data

$$(X_1, Y_1), \ldots, (X_n, Y_n)$$

repeatedly until no more mistakes (i.e. $T \leq n \times (D^2/\gamma^2)$).

Then **final** hyperplane of Perceptron separates the data perfectly, i.e. finds minimum $\widehat{L}_{01}(w_T) = 0$ for

$$\widehat{L}_{01}(w) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{I}\{Y_i \langle w, X_i \rangle \leq 0\}.$$

Empirical Risk Minimization with finite-time convergence. Can we say anything about future performance of w_T?

Consequence for i.i.d. data (II)

Claim:

$$\mathbb{E} L_{01}(w_T) \leq \frac{1}{n+1} \times \mathbb{E}\left[\frac{D^2}{\gamma^2}\right]$$

Proof: Shortcuts: $z = (x, y)$ and $\ell(w, z) = \mathbf{I}\{y \langle w, x\rangle \leq 0\}$. Then

$$\mathbb{E}_{\mathcal{S}}\mathbb{E}_Z \ell(w_T, Z) = \mathbb{E}_{\mathcal{S}, Z_{n+1}} \left[\frac{1}{n+1} \sum_{t=1}^{n+1} \ell(w^{(-t)}, Z_t) \right]$$

where $w^{(-t)}$ is Perceptron's final hyperplane after cycling through data $Z_1, \ldots, Z_{t-1}, Z_{t+1}, \ldots, Z_{n+1}$.

That is, *leave-one-out* is unbiased estimate of expected loss.

Now consider cycling Perceptron on $Z_1, \ldots, Z_{n+1}$ until no more errors. Let $i_1, \ldots, i_m$ be indices on which Perceptron errs in *any* of the cycles. We know $m \leq D^2/\gamma^2$. However, if index $t \notin \{i_1, \ldots, i_m\}$, then whether or not Z_t was included does not matter, and Z_t is correctly classified by $w^{(-t)}$. Claim follows.

A few remarks

- Bayes error $L_{01}(f^*) = 0$ since we assume margin in the distribution P (and, hence, in the data).
- Our assumption on P is not about its parametric or nonparametric form, but rather on what happens at the boundary.
- Curiously, we beat the CLT rate of $1/\sqrt{n}$.

Overparametrization and overfitting

The last lecture will discuss overparametrization and overfitting.

- dimension d never appears in the mistake bound of Perceptron. Hence, we can even take $d = \infty$.
- Perceptron is 1-layer neural network (no hidden layer) with d neurons.
- For $d > n$, any n points in general position can be separated (zero empirical error).

Conclusions:

- More parameters than data does not necessarily contradict good prediction performance ("generalization").
- Perfectly fitting the data does not necessarily contradict good generalization (particularly with no noise).
- Complexity is subtle (e.g. number of parameters vs margin)
- ... however, margin is a strong assumption (more than just no noise)

Outline

Relaxing the assumptions

We proved

$$\mathbb{E}L_{01}(w_T) - L_{01}(f^*) \leq O(1/n)$$

under the assumption that P is linearly separable with margin γ.

The assumption on the probability distribution implies that the Bayes classifier is a linear separator ("realizable" setup).

We may relax the margin assumption, yet still hope to do well with linear separators. That is, we would aim to minimize

$$\mathbb{E}L_{01}(w_T) - L_{01}(w^*)$$

hoping that

$$L_{01}(w^*) - L_{01}(f^*)$$

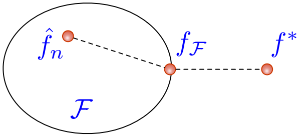
is small, where w^* is the best hyperplane classifier with respect to L_{01}.

More generally, we will work with some class of functions $\mathcal{F}$ that we hope captures well the relationship between X and Y. The choice of $\mathcal{F}$ gives rise to a bias-variance decomposition.

Let $f_{\mathcal{F}} \in \operatorname*{argmin}_{f \in \mathcal{F}} L(f)$ be the function in $\mathcal{F}$ with smallest expected loss.

Bias-Variance Tradeoff

$$L(\widehat{f}_n) - L(f^*) = \underbrace{L(\widehat{f}_n) - \inf_{f\in\mathcal{F}} L(f)}_{\text{Estimation Error}} + \underbrace{\inf_{f\in\mathcal{F}} L(f) - L(f^*)}_{\text{Approximation Error}}$$



Clearly, the two terms are at odds with each other:

- Making $\mathcal{F}$ larger means smaller approximation error but (as we will see) larger estimation error
- Taking a larger sample n means smaller estimation error and has no effect on the approximation error.
- Thus, it makes sense to trade off size of $\mathcal{F}$ and n (*Structural Risk Minimization*, or *Method of Sieves*, or *Model Selection*).

Bias-Variance Tradeoff

In simple problems (e.g. linearly separable data) we might get away with having no bias-variance decomposition (e.g. as was done in the Perceptron case).

However, for more complex problems, our assumptions on P often imply that class containing f^* is too large and the estimation error cannot be controlled. In such cases, we need to produce a "biased" solution in hopes of reducing variance.

Finally, the bias-variance tradeoff need not be in the form we just presented. We will consider a different decomposition in the last lecture.

Outline

Consider the performance of empirical risk minimization:

Choose $\widehat{f}_n \in \mathcal{F}$ to minimize $\widehat{L}(f)$, where $\widehat{L}$ is the *empirical risk*,

$$\widehat{L}(f) = P_n \ell(f(X), Y) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(X_i), Y_i).$$

For pattern classification, this is the proportion of training examples misclassified.

How does the excess risk, $L(\widehat{f}_n) - L(f_{\mathcal{F}})$ behave? We can write

$$L(\widehat{f}_n) - L(f_{\mathcal{F}}) = \left[L(\widehat{f}_n) - \widehat{L}(\widehat{f}_n)\right] + \left[\widehat{L}(\widehat{f}_n) - \widehat{L}(f_{\mathcal{F}})\right] + \left[\widehat{L}(f_{\mathcal{F}}) - L(f_{\mathcal{F}})\right]$$

Therefore,

$$\mathbb{E}L(\widehat{f}_n) - L(f_{\mathcal{F}}) \leq \mathbb{E}\left[L(\widehat{f}_n) - \widehat{L}(\widehat{f}_n)\right]$$

because second term is nonpositive and third is zero in expectation.

The term $L(\widehat{f}_n) - \widehat{L}(\widehat{f}_n)$ is not necessarily zero in expectation (check!). An easy upper bound is

$$L(\widehat{f}_n) - \widehat{L}(\widehat{f}_n) \leq \sup_{f \in \mathcal{F}} \left| L(f) - \widehat{L}(f) \right|,$$

and this motivates the study of uniform laws of large numbers.

Roadmap: study the sup to

- understand when learning is possible,
- understand implications for sample complexity,
- design new algorithms (regularizer arises from upper bound on sup)

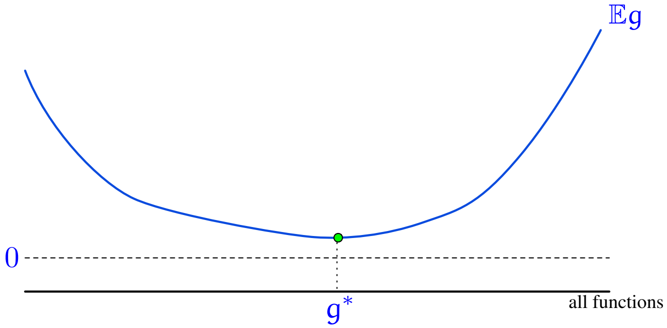
Loss Class

In what follows, we will work with a function class $\mathcal{G}$ on $\mathcal{Z}$, and for the learning application, $\mathcal{G} = \ell \circ \mathcal{F}$:

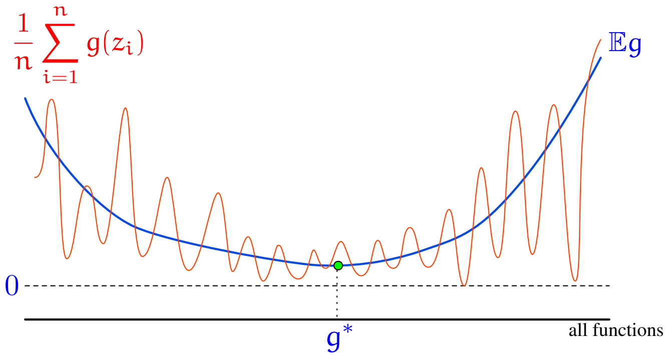
$$\mathcal{G} = \{(x, y) \mapsto \ell(f(x), y) : f \in \mathcal{F}\}$$

and $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$.

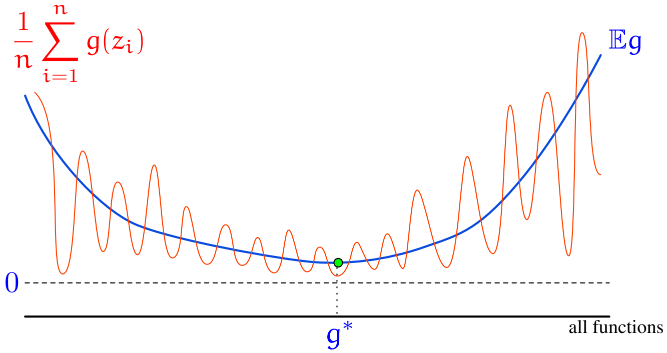
Empirical Process Viewpoint



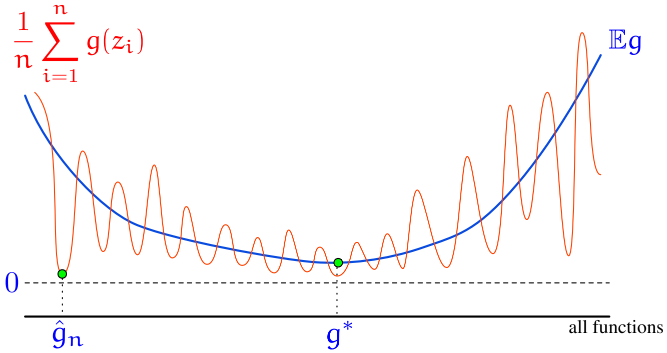
Empirical Process Viewpoint



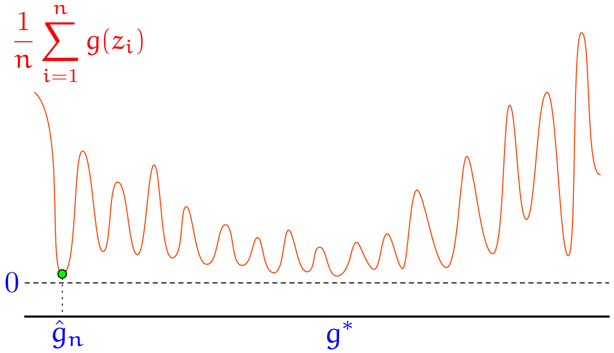
Empirical Process Viewpoint



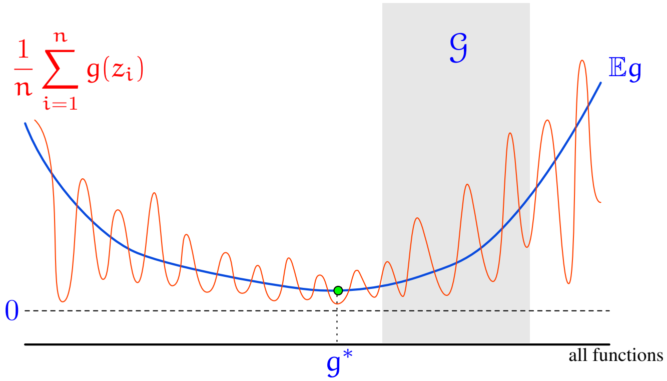
Empirical Process Viewpoint



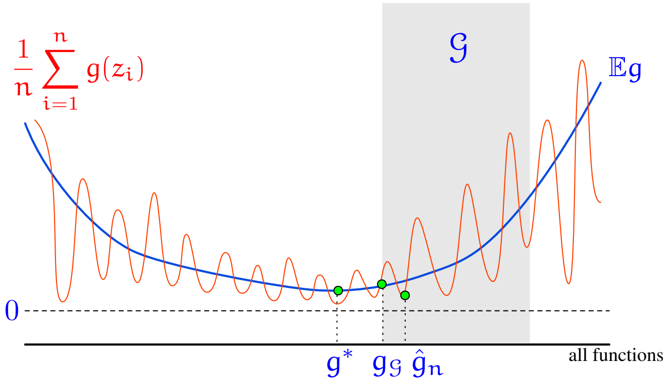
Empirical Process Viewpoint



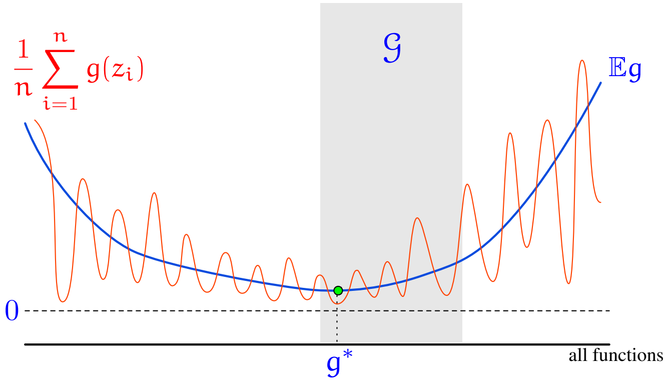
Empirical Process Viewpoint



Empirical Process Viewpoint



Empirical Process Viewpoint



Uniform Laws of Large Numbers

For a class $\mathcal{G}$ of functions $g : \mathcal{Z} \to [0, 1]$, suppose that $Z_1, \ldots, Z_n, Z$ are i.i.d. on $\mathcal{Z}$, and consider

$$U = \sup_{g \in \mathcal{G}} \left| \mathbb{E} g(Z) - \frac{1}{n} \sum_{i=1}^{n} g(Z_i) \right| = \sup_{g \in \mathcal{G}} |Pg - P_n g| =: \| \underbrace{P - P_n}_{\text{emp proc}} \|_{\mathcal{G}}.$$

If U converges to 0, this is called a *uniform law of large numbers*.

Glivenko-Cantelli Classes

Definition.

$\mathcal{G}$ is a **Glivenko-Cantelli class** for P if $\|P_n - P\|_{\mathcal{G}} \xrightarrow{P} 0$.

- P is a distribution on $\mathcal{Z}$,
- $Z_1, \ldots, Z_n$ are drawn i.i.d. from P,
- P_n is the empirical distribution (mass $1/n$ at each of $Z_1, \ldots, Z_n$),
- $\mathcal{G}$ is a set of measurable real-valued functions on $\mathcal{Z}$ with finite expectation under P,
- $P_n - P$ is an **empirical process**, that is, a stochastic process indexed by a class of functions $\mathcal{G}$, and
- $\|P_n - P\|_{\mathcal{G}} := \sup_{g \in \mathcal{G}} |P_n g - Pg|$.

Glivenko-Cantelli Classes

Why 'Glivenko-Cantelli'? An example of a uniform law of large numbers.

Theorem (Glivenko-Cantelli)**.**

$$\|F_n - F\|_\infty \overset{as}{\to} 0.$$

Here, F is a cumulative distribution function, F_n is the empirical cumulative distribution function,

$$F_n(t) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{Z_i \le t\},$$

where $Z_1, \ldots, Z_n$ are i.i.d. with distribution F, and $\|F - G\|_\infty = \sup_t |F(t) - G(t)|$.

Theorem (Glivenko-Cantelli)**.**

$\|P_n - P\|_{\mathcal{G}} \overset{as}{\to} 0$, for $\mathcal{G} = \{z \mapsto \mathbb{I}\{z \le \theta\} : \theta \in \mathbb{R}\}$.

Glivenko-Cantelli Classes

Not all $\mathcal{G}$ are Glivenko-Cantelli classes. For instance,

$$\mathcal{G} = \left\{ \mathbf{I}\{z \in S\} : S \subset \mathbb{R}, |S| < \infty \right\}.$$

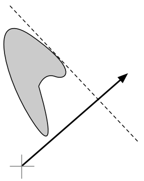
Then for a continuous distribution P, $Pg = 0$ for any $g \in \mathcal{G}$, but $\sup_{g \in \mathcal{G}} P_n g = 1$ for all n. So although $P_n g \overset{as}{\to} Pg$ for all $g \in \mathcal{G}$, this convergence is not uniform over $\mathcal{G}$. $\mathcal{G}$ is **too large**.

In general, how do we decide whether $\mathcal{G}$ is "too large"?

Outline

Rademacher Averages

Given a set $H \subseteq [-1, 1]^n$, measure its "size" by average length of projection onto random direction.



Rademacher process indexed by vector $h \in H$:

$$\widehat{R}_n(h) = \frac{1}{n} \langle \epsilon, h \rangle$$

where $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$ is vector of i.i.d. Rademacher random variables (symmetric ± 1).

Rademacher averages: expected maximum of the process

$$\mathbb{E}_\epsilon \left\| \widehat{R}_n \right\|_H = \mathbb{E} \sup_{h \in H} \left| \frac{1}{n} \langle \epsilon, h \rangle \right|$$

Examples

If $H = \{h_0\}$,

$$\mathbb{E}\left\|\widehat{R}_n\right\|_H = O(n^{-1/2}).$$

If $H = \{-1, 1\}^n$,

$$\mathbb{E}\left\|\widehat{R}_n\right\|_H = 1.$$

If $v + c\{-1, 1\}^n \subseteq H$, then

$$\mathbb{E}\left\|\widehat{R}_n\right\|_H \geq c - o(1)$$

How about $H = \{-\mathbf{1}, \mathbf{1}\}$ where $\mathbf{1}$ is a vector of 1's?

Examples

If H is finite,

$$\mathbb{E}\left\|\widehat{R}_n\right\|_H \leq c\sqrt{\frac{\log|H|}{n}}$$

for some constant c.

This bound can be loose, as it does not take into account "overlaps"/correlations between vectors.

Examples

Let B_p^n be a unit ball in $\mathbb{R}^n$:

$$\mathrm{B}_p^n = \{x \in \mathbb{R}^n : \|x\|_p \leq 1\}, \quad \|x\|_p = \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}.$$

For $H = \mathrm{B}_2^n$

$$\frac{1}{n}\mathbb{E} \max_{\|g\|_2 \leq 1} |\langle \epsilon, g \rangle| = \frac{1}{n}\mathbb{E}\,\|\epsilon\|_2 = \frac{1}{\sqrt{n}}$$

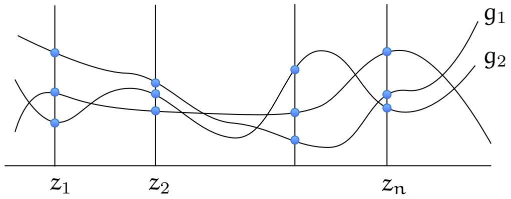
On the other hand, Rademacher averages for B_1^n are $\frac{1}{n}$.

Rademacher Averages

What do these Rademacher averages have to do with our problem of bounding uniform deviations?

Suppose we take

$$H = \mathcal{G}(Z_1^n) = \{(g(Z_1), \ldots, g(Z_n)) : g \in \mathcal{G}\} \subset \mathbb{R}^n$$



Rademacher Averages

Then, conditionally on Z_1^n,

$$\mathbb{E}_\epsilon \left\| \widehat{R}_n \right\|_H = \mathbb{E}_\epsilon \sup_{g \in \mathcal{G}} \left| \frac{1}{n} \sum_{i=1}^n \epsilon_i g(Z_i) \right|$$

is a data-dependent quantity (hence the "hat") that measures complexity of $\mathcal{G}$ projected onto data.

Also define the Rademacher process R_n (without the "hat")

$$R_n(g) = \frac{1}{n} \sum_{i=1}^n \epsilon_i g(Z_i),$$

and **Rademacher complexity** of $\mathcal{G}$ as

$$\mathbb{E}\|R_n\|_{\mathcal{G}} = \mathbb{E}_{Z_1^n} \mathbb{E}_\epsilon \left\| \widehat{R}_n \right\|_{\mathcal{G}(Z_1^n)}.$$

Uniform Laws and Rademacher Complexity

We'll look at a proof of a uniform law of large numbers that involves two steps:

1. Symmetrization, which bounds $\mathbb{E}\|P - P_n\|_{\mathcal{G}}$ in terms of the Rademacher complexity of F, $\mathbb{E}\|R_n\|_{\mathcal{G}}$.
2. Both $\|P - P_n\|_{\mathcal{G}}$ and $\|R_n\|_{\mathcal{G}}$ are concentrated about their expectation.

Uniform Laws and Rademacher Complexity

Theorem.

For any $\mathcal{G}$,

$$\mathbb{E}\|P - P_n\|_{\mathcal{G}} \le 2\mathbb{E}\|R_n\|_{\mathcal{G}}$$

If $\mathcal{G} \subset [0,1]^{\mathcal{Z}}$, then

$$\frac{1}{2}\mathbb{E}\|R_n\|_{\mathcal{G}} - \sqrt{\frac{\log 2}{2n}} \le \mathbb{E}\|P - P_n\|_{\mathcal{G}} \le 2\mathbb{E}\|R_n\|_{\mathcal{G}},$$

and, with probability at least $1 - 2\exp(-2\epsilon^2 n)$,

$$\mathbb{E}\|P - P_n\|_{\mathcal{G}} - \epsilon \le \|P - P_n\|_{\mathcal{G}} \le \mathbb{E}\|P - P_n\|_{\mathcal{G}} + \epsilon.$$

Thus, $\mathbb{E}\|R_n\|_{\mathcal{G}} \to 0$ iff $\|P - P_n\|_{\mathcal{G}} \xrightarrow{as} 0$.

That is, the supremum of the empirical process $P - P_n$ is concentrated about its expectation, and its expectation is about the same as the expected sup of the Rademacher process R_n.

Uniform Laws and Rademacher Complexity

The first step is to symmetrize by replacing Pg by $P'_n g = \frac{1}{n}\sum_{i=1}^n g(Z'_i)$. In particular, we have

$$\mathbb{E}\|P - P_n\|_{\mathcal{G}} = \mathbb{E}\sup_{g\in\mathcal{G}}\left|\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n (g(Z'_i) - g(Z_i))\middle| Z_1^n\right]\right|$$

Uniform Laws and Rademacher Complexity

The first step is to symmetrize by replacing Pg by $P'_n g = \frac{1}{n}\sum_{i=1}^n g(Z'_i)$. In particular, we have

$$\begin{aligned}\mathbb{E}\|P - P_n\|_{\mathcal{G}} &= \mathbb{E}\sup_{g\in\mathcal{G}}\left|\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n (g(Z'_i) - g(Z_i))\middle| Z_1^n\right]\right| \\ &\leq \mathbb{E}\mathbb{E}\left[\sup_{g\in\mathcal{G}}\left|\frac{1}{n}\sum_{i=1}^n (g(Z'_i) - g(Z_i))\right|\middle| Z_1^n\right]\end{aligned}$$

Uniform Laws and Rademacher Complexity

The first step is to symmetrize by replacing Pg by $P'_n g = \frac{1}{n}\sum_{i=1}^n g(Z'_i)$. In particular, we have

$$\begin{aligned}
\mathbb{E}\|P - P_n\|_{\mathcal{G}} &= \mathbb{E}\sup_{g\in\mathcal{G}}\left|\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n (g(Z'_i) - g(Z_i))\middle| Z_1^n\right]\right| \\
&\le \mathbb{E}\mathbb{E}\left[\sup_{g\in\mathcal{G}}\left|\frac{1}{n}\sum_{i=1}^n (g(Z'_i) - g(Z_i))\right|\middle| Z_1^n\right] \\
&= \mathbb{E}\sup_{g\in\mathcal{G}}\left|\frac{1}{n}\sum_{i=1}^n (g(Z'_i) - g(Z_i))\right| = \mathbb{E}\|P'_n - P_n\|_{\mathcal{G}}.
\end{aligned}$$

Uniform Laws and Rademacher Complexity

Another symmetrization: for any $\epsilon_i \in \{\pm 1\}$,

$$\begin{aligned}\mathbb{E}\|P_n' - P_n\|_{\mathcal{G}} &= \mathbb{E}\sup_{g\in\mathcal{G}}\left|\frac{1}{n}\sum_{i=1}^{n}(g(Z_i') - g(Z_i))\right| \\ &= \mathbb{E}\sup_{g\in\mathcal{G}}\left|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i(g(Z_i') - g(Z_i))\right|,\end{aligned}$$

This follows from the fact that Z_i and Z_i' are i.i.d., and so the distribution of the supremum is unchanged when we swap them. And so in particular the expectation of the supremum is unchanged. And since this is true for any ϵ_i, we can take the expectation over any random choice of the ϵ_i. We'll pick them independently and uniformly.

Uniform Laws and Rademacher Complexity

$$\begin{aligned}
&\mathbb{E} \sup_{g \in \mathcal{G}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i (g(Z_i') - g(Z_i)) \right| \\
&\leq \mathbb{E} \sup_{g \in \mathcal{G}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i g(Z_i') \right| + \sup_{g \in \mathcal{G}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i g(Z_i) \right| \\
&= \underbrace{2\, \mathbb{E} \sup_{g \in \mathcal{G}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i g(Z_i) \right|}_{\text{Rademacher complexity}} = 2\mathbb{E}\|R_n\|_{\mathcal{G}},
\end{aligned}$$

where R_n is the *Rademacher process* $R_n(g) = (1/n) \sum_{i=1}^{n} \epsilon_i g(Z_i)$.

Uniform Laws and Rademacher Complexity

The second inequality (*desymmetrization*) follows from:

$$\begin{aligned}
\mathbb{E}\|R_n\|_{\mathcal{G}} &\le \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\left(g(Z_i)-\mathbb{E}g(Z_i)\right)\right\|_{\mathcal{G}} + \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\mathbb{E}g(Z_i)\right\|_{\mathcal{G}} \\
&\le \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\left(g(Z_i)-g(Z_i')\right)\right\|_{\mathcal{G}} + \|P\|_{\mathcal{G}}\,\mathbb{E}\left|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\right| \\
&= \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\left(g(Z_i)-\mathbb{E}g(Z_i)+\mathbb{E}g(Z_i')-g(Z_i')\right)\right\|_{\mathcal{G}} \\
&\qquad + \|P\|_{\mathcal{G}}\,\mathbb{E}\left|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\right| \\
&\le 2\mathbb{E}\,\|P_n-P\|_{\mathcal{G}} + \sqrt{\frac{2\log 2}{n}}.
\end{aligned}$$

Uniform Laws and Rademacher Complexity

Theorem.

For any $\mathcal{G}$,

$$\mathbb{E}\|P - P_n\|_{\mathcal{G}} \leq 2\mathbb{E}\|R_n\|_{\mathcal{G}}$$

If $\mathcal{G} \subset [0,1]^{\mathcal{Z}}$, then

$$\frac{1}{2}\mathbb{E}\|R_n\|_{\mathcal{G}} - \sqrt{\frac{\log 2}{2n}} \leq \mathbb{E}\|P - P_n\|_{\mathcal{G}} \leq 2\mathbb{E}\|R_n\|_{\mathcal{G}},$$

and, with probability at least $1 - 2\exp(-2\epsilon^2 n)$,

$$\mathbb{E}\|P - P_n\|_{\mathcal{G}} - \epsilon \leq \|P - P_n\|_{\mathcal{G}} \leq \mathbb{E}\|P - P_n\|_{\mathcal{G}} + \epsilon.$$

Thus, $\mathbb{E}\|R_n\|_{\mathcal{G}} \to 0$ iff $\|P - P_n\|_{\mathcal{G}} \overset{as}{\to} 0$.

That is, the supremum of the empirical process $P - P_n$ is concentrated about its expectation, and its expectation is about the same as the expected sup of the Rademacher process R_n.

Back to the prediction problem

Recall: for prediction problem, we have $\mathcal{G} = \ell \circ \mathcal{F}$ for some loss function ℓ and a set of functions $\mathcal{F}$.

We found that $\mathbb{E}\left\|P - P_n\right\|_{\ell\circ\mathcal{F}}$ can be related to $\mathbb{E}\left\|R_n\right\|_{\ell\circ\mathcal{F}}$. Question: can we get $\mathbb{E}\left\|R_n\right\|_{\mathcal{F}}$ instead?

For classification, this is easy. Suppose $\mathcal{F} \subseteq \{\pm 1\}^{\mathcal{X}}$, $Y \in \{\pm 1\}$, and $\ell(y, y') = \mathbb{1}\{y \neq y'\} = (1 - yy')/2$. Then

$$
\begin{aligned}
\mathbb{E}\left\|R_n\right\|_{\ell\circ\mathcal{F}} &= \mathbb{E}\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i \ell(y_i, f(x_i))\right| \\
&= \frac{1}{2}\mathbb{E}\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i + \frac{1}{n}\sum_{i=1}^{n}-\epsilon_i y_i f(x_i)\right| \\
&\leq O\left(\frac{1}{\sqrt{n}}\right) + \frac{1}{2}\mathbb{E}\left\|R_n\right\|_{\mathcal{F}}.
\end{aligned}
$$

Other loss functions?

Outline

Rademacher Complexity: Structural Results

Theorem.

Subsets	$\mathcal{F} \subseteq \mathcal{G}$ implies $\|R_n\|_{\mathcal{F}} \leq \|R_n\|_{\mathcal{G}}$.
Scaling	$\|R_n\|_{c\mathcal{F}} = \|c\|\|R_n\|_{\mathcal{F}}$.
Plus Constant	For $\|g(X)\| \leq 1$, $\left\|\mathbb{E}\|R_n\|_{\mathcal{F}+g} - \mathbb{E}\|R_n\|_{\mathcal{F}}\right\| \leq \sqrt{2\log 2/n}$.
Convex Hull	$\|R_n\|_{\mathrm{co}\mathcal{F}} = \|R_n\|_{\mathcal{F}}$, where $\mathrm{co}\mathcal{F}$ is the convex hull of $\mathcal{F}$.
Contraction	If $\ell : \mathbb{R} \times \mathbb{R} \to [-1,1]$ has $\hat{y} \mapsto \ell(\hat{y}, y)$ 1-Lipschitz for all y, then for $\ell \circ \mathcal{F} = \{(x,y) \mapsto \ell(f(x), y)\}$, $\mathbb{E}\|R_n\|_{\ell\circ\mathcal{F}} \leq 2\mathbb{E}\|R_n\|_{\mathcal{F}} + c/\sqrt{n}$.

Outline

Deep Networks

Deep compositions of nonlinear functions

$$h = h_m \circ h_{m-1} \circ \cdots \circ h_1$$

Deep Networks

Deep compositions of nonlinear functions

$$h = h_m \circ h_{m-1} \circ \cdots \circ h_1$$

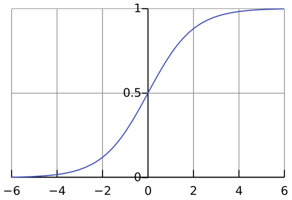
e.g., $$h_i : x \mapsto \sigma(W_i x)$$
$$\sigma(v)_i = \frac{1}{1 + \exp(-v_i)},$$

Deep Networks

Deep compositions of nonlinear functions

$$h = h_m \circ h_{m-1} \circ \cdots \circ h_1$$

$$\text{e.g.,} \qquad h_i : x \mapsto \sigma(W_i x)$$
$$\sigma(v)_i = \frac{1}{1 + \exp(-v_i)},$$



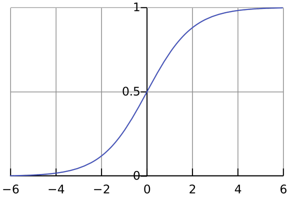
Deep Networks

Deep compositions of nonlinear functions

$$h = h_m \circ h_{m-1} \circ \cdots \circ h_1$$

e.g., $h_i : x \mapsto \sigma(W_i x)$ $\qquad$ $h_i : x \mapsto r(W_i x)$

$$\sigma(v)_i = \frac{1}{1 + \exp(-v_i)}, \qquad r(v)_i = \max\{0, v_i\}$$



Deep Networks

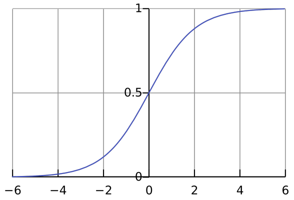
Deep compositions of nonlinear functions

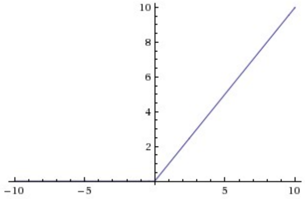
$$h = h_m \circ h_{m-1} \circ \cdots \circ h_1$$

e.g.,

$$h_i : x \mapsto \sigma(W_i x) \qquad\qquad h_i : x \mapsto r(W_i x)$$

$$\sigma(v)_i = \frac{1}{1+\exp(-v_i)}, \qquad\qquad r(v)_i = \max\{0, v_i\}$$





Rademacher Complexity of Sigmoid Networks

Theorem.

Consider the following class $\mathcal{F}_{B,d}$ of two-layer neural networks on $\mathbb{R}^d$:

$$\mathcal{F}_{B,d} = \left\{ x \mapsto \sum_{i=1}^{k} w_i \sigma\left(v_i^T x\right) : \|w\|_1 \leq B,\ \|v_i\|_1 \leq B,\ k \geq 1 \right\},$$

where $B > 0$ and the nonlinear function $\sigma : \mathbb{R} \to \mathbb{R}$ satisfies the Lipschitz condition, $|\sigma(a) - \sigma(b)| \leq |a - b|$, and $\sigma(0) = 0$. Suppose that the distribution is such that $\|X\|_\infty \leq 1$ a.s. Then

$$\mathbb{E}\|R_n\|_{\mathcal{F}_{B,d}} \leq 2B^2 \sqrt{\frac{2 \log 2d}{n}},$$

where d is the dimension of the input space, $\mathcal{X} = \mathbb{R}^d$.

Rademacher Complexity of Sigmoid Networks: Proof

Define $$\mathcal{G} := \{(x_1, \ldots, x_d) \mapsto x_j : 1 \le j \le d\},$$

$$\mathcal{V}_B := \left\{ x \mapsto v'x \;:\; \|v\|_1 = \sum_{i=1}^{d} |v_i| \le B \right\}$$
$$= B\mathrm{co}\left(\mathcal{G} \cup -\mathcal{G}\right),$$

with $$\mathrm{co}(\mathcal{F}) := \left\{ \sum_{i=1}^{k} \alpha_i f_i \;:\; k \ge 1, \alpha_i \ge 0, \|\alpha\|_1 = 1, f_i \in \mathcal{F} \right\}.$$

Rademacher Complexity of Sigmoid Networks: Proof

Define $\mathcal{G} := \{(x_1, \ldots, x_d) \mapsto x_j : 1 \leq j \leq d\},$

$$\mathcal{V}_B := \left\{ x \mapsto v'x \;:\; \|v\|_1 = \sum_{i=1}^{d} |v_i| \leq B \right\}$$
$$= B\mathrm{co}\left(\mathcal{G} \cup -\mathcal{G}\right),$$

with $$\mathrm{co}(\mathcal{F}) := \left\{ \sum_{i=1}^{k} \alpha_i f_i \;:\; k \geq 1, \alpha_i \geq 0, \|\alpha\|_1 = 1, f_i \in \mathcal{F} \right\}.$$

Then $$\mathcal{F}_{B,d} = \left\{ x \mapsto \sum_{i=1}^{k} w_i \sigma(v_i(x)) \mid k \geq 1, \sum_{i=1}^{k} w_i \leq B, \; v_i \in \mathcal{V}_B \right\}$$
$$= B\mathrm{co}\left(\sigma \circ \mathcal{V}_B \cup -\sigma \circ \mathcal{V}_B\right).$$

Rademacher Complexity of Sigmoid Networks: Proof

$$
\begin{aligned}
\mathbb{E}\|R_n\|_{\mathcal{F}_{B,d}} &= \mathbb{E}\|R_n\|_{B\,\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)} \\
&= B\mathbb{E}\|R_n\|_{\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)}
\end{aligned}
$$

Rademacher Complexity of Sigmoid Networks: Proof

$$\begin{aligned}
\mathbb{E}\|R_n\|_{\mathcal{F}_{B,d}} &= \mathbb{E}\|R_n\|_{B\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)} \\
&= B\mathbb{E}\|R_n\|_{\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)} \\
&= B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B}
\end{aligned}$$

Rademacher Complexity of Sigmoid Networks: Proof

$$\begin{aligned}
\mathbb{E}\|R_n\|_{\mathcal{F}_{B,d}} &= \mathbb{E}\|R_n\|_{B\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)} \\
&= B\mathbb{E}\|R_n\|_{\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)} \\
&= B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B} \\
&\leq B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B} + B\mathbb{E}\|R_n\|_{-\sigma\circ\mathcal{V}_B}
\end{aligned}$$

Rademacher Complexity of Sigmoid Networks: Proof

$$\begin{aligned}
\mathbb{E}\|R_n\|_{\mathcal{F}_{B,d}} &= \mathbb{E}\|R_n\|_{B\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)} \\
&= B\mathbb{E}\|R_n\|_{\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)} \\
&= B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B} \\
&\leq B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B} + B\mathbb{E}\|R_n\|_{-\sigma\circ\mathcal{V}_B} \\
&= 2B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B}
\end{aligned}$$

Rademacher Complexity of Sigmoid Networks: Proof

$$\begin{aligned}
\mathbb{E}\|R_n\|_{\mathcal{F}_{B,d}} &= \mathbb{E}\|R_n\|_{B\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)} \\
&= B\mathbb{E}\|R_n\|_{\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)} \\
&= B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B} \\
&\leq B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B} + B\mathbb{E}\|R_n\|_{-\sigma\circ\mathcal{V}_B} \\
&= 2B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B} \\
&\leq 2B\mathbb{E}\|R_n\|_{\mathcal{V}_B}
\end{aligned}$$

Rademacher Complexity of Sigmoid Networks: Proof

$$\begin{aligned}
\mathbb{E}\|R_n\|_{\mathcal{F}_{B,d}} &= \mathbb{E}\|R_n\|_{B\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)}\\
&= B\mathbb{E}\|R_n\|_{\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)}\\
&= B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B}\\
&\leq B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B} + B\mathbb{E}\|R_n\|_{-\sigma\circ\mathcal{V}_B}\\
&= 2B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B}\\
&\leq 2B\mathbb{E}\|R_n\|_{\mathcal{V}_B}\\
&= 2B\mathbb{E}\|R_n\|_{B\mathrm{co}(\mathcal{G}\cup-\mathcal{G})}
\end{aligned}$$

Rademacher Complexity of Sigmoid Networks: Proof

$$\begin{aligned}
\mathbb{E}\|R_n\|_{\mathcal{F}_{B,d}} &= \mathbb{E}\|R_n\|_{B\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)} \\
&= B\mathbb{E}\|R_n\|_{\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)} \\
&= B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B} \\
&\leq B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B} + B\mathbb{E}\|R_n\|_{-\sigma\circ\mathcal{V}_B} \\
&= 2B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B} \\
&\leq 2B\mathbb{E}\|R_n\|_{\mathcal{V}_B} \\
&= 2B\mathbb{E}\|R_n\|_{B\mathrm{co}(\mathcal{G}\cup-\mathcal{G})} \\
&= 2B^2\mathbb{E}\|R_n\|_{\mathcal{G}\cup-\mathcal{G}}
\end{aligned}$$

Rademacher Complexity of Sigmoid Networks: Proof

$$\begin{aligned}
\mathbb{E}\|R_n\|_{\mathcal{F}_{B,d}} &= \mathbb{E}\|R_n\|_{B\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)} \\
&= B\mathbb{E}\|R_n\|_{\mathrm{co}(\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B)} \\
&= B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B\cup-\sigma\circ\mathcal{V}_B} \\
&\le B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B} + B\mathbb{E}\|R_n\|_{-\sigma\circ\mathcal{V}_B} \\
&= 2B\mathbb{E}\|R_n\|_{\sigma\circ\mathcal{V}_B} \\
&\le 2B\mathbb{E}\|R_n\|_{\mathcal{V}_B} \\
&= 2B\mathbb{E}\|R_n\|_{B\mathrm{co}(\mathcal{G}\cup-\mathcal{G})} \\
&= 2B^2\mathbb{E}\|R_n\|_{\mathcal{G}\cup-\mathcal{G}} \\
&\le 2B^2\sqrt{\frac{2\log(2d)}{n}}.
\end{aligned}$$

Rademacher Complexity of Sigmoid Networks: Proof

The final step involved the following result.

Lemma.

For $f \in \mathcal{F}$ satisfying $|f(x)| \le 1$,

$$\mathbb{E}\|R_n\|_{\mathcal{F}} \le \mathbb{E}\sqrt{\frac{2\log(2|\mathcal{F}(X_1^n)|)}{n}}.$$

Outline

Rademacher Complexity of ReLU Networks

Using the positive homogeneity property of the ReLU nonlinearity (that is, for all $\alpha \geq 0$ and $x \in \mathbb{R}$, $\sigma(\alpha x) = \alpha\sigma(x)$) gives a simple argument to bound the Rademacher complexity.

Theorem.

For a training set $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$ with $\|X_i\| \leq 1$ a.s.,

$$\mathbb{E}\,\|R_n\|_{\mathcal{F}_{F,B}} \leq \frac{(2B)^L}{\sqrt{n}},$$

where $f \in \mathcal{F}_{F,B}$ is an L-layer network of the form

$$\mathcal{F}_{F,B} := W_L\sigma(W_{L-1}\cdots\sigma(W_1 x)\cdots),$$

σ is 1-Lipschitz, positive homogeneous (that is, for all $\alpha \geq 0$ and $x \in \mathbb{R}$, $\sigma(\alpha x) = \alpha\sigma(x)$), and applied componentwise, and $\|W_i\|_F \leq B$. (W_L is a row vector.)

ReLU Networks: Proof

(Write $\mathbb{E}_\epsilon$ as the conditional expectation given the data.)

Lemma.

$$\mathbb{E}_\epsilon \sup_{f\in\mathcal{F},\|W\|_F\le B} \frac{1}{n}\left\|\sum_{i=1}^{n}\epsilon_i\sigma(Wf(X_i))\right\|_2 \le 2B\mathbb{E}_\epsilon \sup_{f\in\mathcal{F}} \frac{1}{n}\left\|\sum_{i=1}^{n}\epsilon_i f(X_i)\right\|_2 .$$

Iterating this and using Jensen's inequality proves the theorem.

ReLU Networks: Proof

For $W^\top = (w_1 \cdots w_k)$, we use positive homogeneity:

$$\begin{aligned}
\left\| \sum_{i=1}^n \epsilon_i \sigma(Wf(x_i)) \right\|^2 &= \sum_{j=1}^k \left(\sum_{i=1}^n \epsilon_i \sigma(w_j^\top f(x_i)) \right)^2 \\
&= \sum_{j=1}^k \|w_j\|^2 \left(\sum_{i=1}^n \epsilon_i \sigma\left(\frac{w_j^\top}{\|w_j\|} f(x_i) \right) \right)^2 .
\end{aligned}$$

ReLU Networks: Proof

$$\begin{aligned}
&\sup_{\|W\|_F \leq B} \sum_{j=1}^{k} \|w_j\|^2 \left(\sum_{i=1}^{n} \epsilon_i \sigma \left(\frac{w_j^\top}{\|w_j\|} f(x_i) \right) \right)^2 \\
&= \sup_{\|w_j\|=1; \|\alpha\|_1 \leq B^2} \sum_{j=1}^{k} \alpha_j \left(\sum_{i=1}^{n} \epsilon_i \sigma \left(w_j^\top f(x_i) \right) \right)^2 \\
&= B^2 \sup_{\|w\|=1} \left(\sum_{i=1}^{n} \epsilon_i \sigma \left(w^\top f(x_i) \right) \right)^2,
\end{aligned}$$

then apply (Contraction) and Cauchy-Schwartz inequalities.

Outline

Rademacher Complexity and Covering Numbers

Definition.

An ϵ-cover of a subset T of a metric space (S, d) is a set $\hat{T} \subset T$ such that for each $t \in T$ there is a $\hat{t} \in \hat{T}$ such that $d(t, \hat{t}) \leq \epsilon$. The ϵ-covering number of T is

$$\mathcal{N}(\epsilon, T, d) = \min\{|\hat{T}| : \hat{T} \text{ is an } \epsilon\text{-cover of } T\}.$$

Intuition: A k-dimensional set T has $\mathcal{N}(\epsilon, T, k) = \Theta(1/\epsilon^k)$.
Example: $([0,1]^k, l_\infty)$.

Rademacher Complexity and Covering Numbers

Theorem.

For any $\mathcal{F}$ and any $X_1, \ldots, X_n$,

$$\mathbb{E}_\epsilon \| \widehat{R}_n \|_{\mathcal{F}(X_1^n)} \leq c \int_0^\infty \sqrt{\frac{\ln \mathcal{N}(\epsilon, \mathcal{F}(X_1^n), \ell_2)}{n}} \, d\epsilon.$$

and

$$\mathbb{E}_\epsilon \| \widehat{R}_n \|_{\mathcal{F}(X_1^n)} \geq \frac{1}{c \log n} \sup_{\alpha > 0} \alpha \sqrt{\frac{\ln \mathcal{N}(\alpha, \mathcal{F}(X_1^n), \ell_2)}{n}}.$$

Covering numbers and smoothly parameterized functions

Lemma.

Let F be a parameterized class of functions,

$$\mathcal{F} = \{f(\theta, \cdot) : \theta \in \Theta\}.$$

Let $\|\cdot\|_\Theta$ be a norm on Θ and let $\|\cdot\|_\mathcal{F}$ be a norm on F. Suppose that the mapping $\theta \mapsto f(\theta, \cdot)$ is L-Lipschitz, that is,

$$\|f(\theta, \cdot) - f(\theta', \cdot)\|_\mathcal{F} \leq L\|\theta - \theta'\|_\Theta.$$

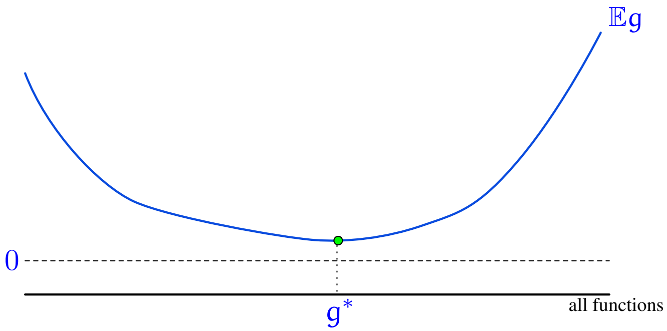
Then $\mathcal{N}(\epsilon, \mathcal{F}, \|\cdot\|_\mathcal{F}) \leq \mathcal{N}(\epsilon/L, \Theta, \|\cdot\|_\Theta)$.

Covering numbers and smoothly parameterized functions

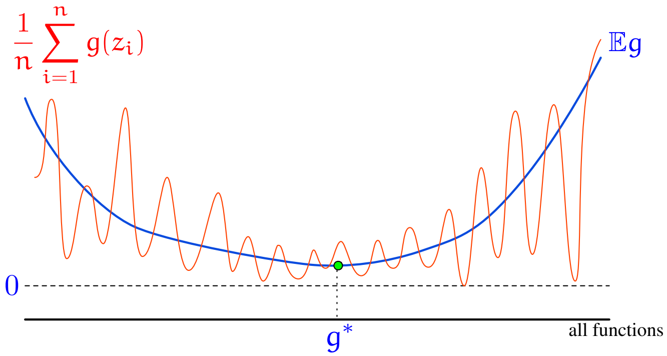
A Lipschitz parameterization allows us to translate a cover of the parameter space into a cover of the function space.

Example: If $\mathcal{F}$ is smoothly parameterized by (a compact set of) k parameters, then $\mathcal{N}(\epsilon, \mathcal{F}) = O(1/\epsilon^k)$.

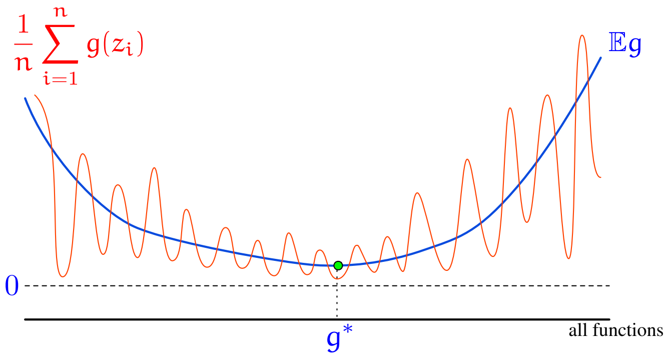
Recap



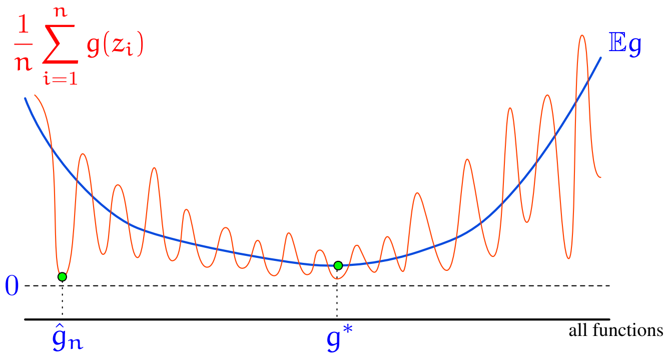
Recap



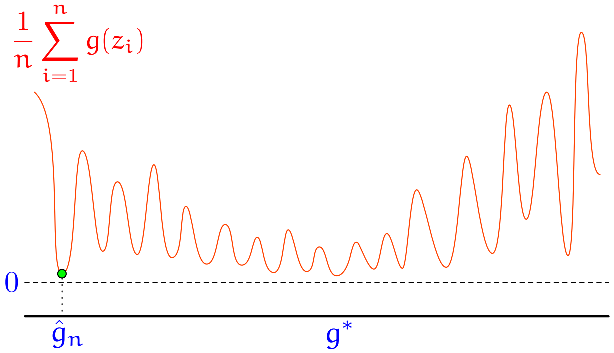
Recap



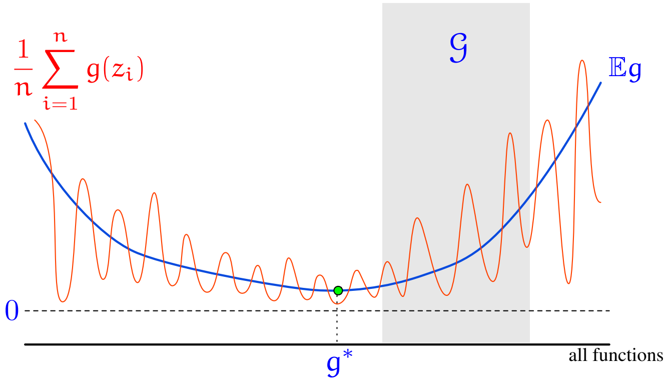
Recap



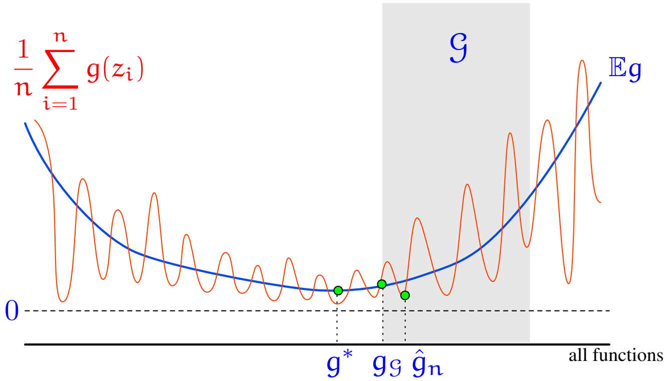
Recap



Recap

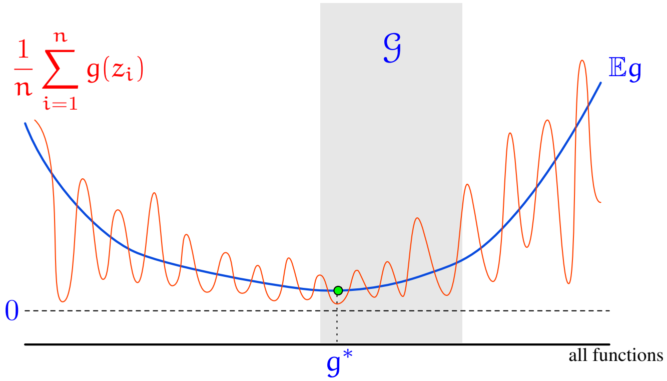


Recap



$\frac{1}{n}\sum_{i=1}^{n} g(z_i)$
$\mathcal{G}$
$\mathbb{E}g$
0
g^*
$g_{\mathcal{G}}$
$\hat{g}_n$
all functions

Recap



Recap

$$\mathbb{E}\left\|P - P_n\right\|_{\mathcal{F}} \asymp \mathbb{E}\left\|R_n\right\|_{\mathcal{F}}$$

Advantage of dealing with Rademacher process:

- often easier to analyze
- can work conditionally on the data

Example: $\mathcal{F} = \{x \mapsto \langle w, x\rangle : \|w\| \le 1\}$, $\|x\| \le 1$. Then

$$\mathbb{E}_\epsilon \left\|\widehat{R}_n\right\|_{\mathcal{F}} = \mathbb{E}_\epsilon \sup_{\|w\|\le 1} \left|\frac{1}{n}\sum_{i=1}^n \epsilon_i \langle w, x_i\rangle\right| = \mathbb{E}_\epsilon \sup_{\|w\|\le 1} \left|\left\langle w, \frac{1}{n}\sum_{i=1}^n \epsilon_i x_i\right\rangle\right|$$

which is equal to

$$\mathbb{E}_\epsilon \left\|\frac{1}{n}\sum_{i=1}^n \epsilon_i x_i\right\| \le \frac{\sqrt{\sum_{i=1}^n \|x_i\|^2}}{n} = \frac{\sqrt{\mathrm{tr}(XX^\top)}}{n}$$

Outline

Classification and Rademacher Complexity

Recall:

For $\mathcal{F} \subseteq \{\pm 1\}^{\mathcal{X}}$, $Y \in \{\pm 1\}$, and $\ell(y, y') = \mathbb{1}\{y \neq y'\} = (1 - yy')/2$,

$$\mathbb{E}\, \|R_n\|_{\ell \circ \mathcal{F}} \leq \frac{1}{2} \mathbb{E}\, \|R_n\|_{\mathcal{F}} + O\left(\frac{1}{\sqrt{n}}\right).$$

How do we get a handle on Rademacher complexity of a set $\mathcal{F}$ of classifiers? Recall that we are looking at "size" of $\mathcal{F}$ projected onto data:

$$\mathcal{F}(x_1^n) = \{(f(x_1), \ldots, f(x_n)) : f \in \mathcal{F}\} \subseteq \{-1, 1\}^n$$

Controlling Rademacher Complexity: Growth Function

Example: For the class of step functions, $\mathcal{F} = \{x \mapsto 1\{x \leq \alpha\} : \alpha \in \mathbb{R}\}$, the set of restrictions,

$$\mathcal{F}(x_1^n) = \{(f(x_1), \ldots, f(x_n)) : f \in \mathcal{F}\}$$

is always small: $|\mathcal{F}(x_1^n)| \leq n + 1$. So $\mathbb{E}\|R_n\|_{\mathcal{F}} \leq \sqrt{\frac{2 \log 2(n+1)}{n}}$.

Controlling Rademacher Complexity: Growth Function

Example: For the class of step functions, $\mathcal{F} = \{x \mapsto \mathbf{1}\{x \leq \alpha\} : \alpha \in \mathbb{R}\}$, the set of restrictions,

$$\mathcal{F}(x_1^n) = \{(f(x_1), \ldots, f(x_n)) : f \in \mathcal{F}\}$$

is always small: $|\mathcal{F}(x_1^n)| \leq n+1$. So $\mathbb{E}\|R_n\|_{\mathcal{F}} \leq \sqrt{\frac{2\log 2(n+1)}{n}}$.

Example: If $\mathcal{F}$ is parameterized by k bits, $\mathcal{F} = \left\{x \mapsto f(x,\theta) : \theta \in \{0,1\}^k\right\}$ for some $f : \mathcal{X} \times \{0,1\}^k \to [0,1]$,

$$|\mathcal{F}(x_1^n)| \leq 2^k,$$

$$\mathbb{E}\|R_n\|_{\mathcal{F}} \leq \sqrt{\frac{2(k+1)\log 2}{n}}.$$

Growth Function

Definition.

For a class $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$, the **growth function** is

$$\Pi_{\mathcal{F}}(n) = \max\{|\mathcal{F}(x_1^n)| : x_1, \ldots, x_n \in \mathcal{X}\}.$$

Growth Function

Definition.

For a class $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$, the **growth function** is

$$\Pi_{\mathcal{F}}(n) = \max\{|\mathcal{F}(x_1^n)| : x_1, \ldots, x_n \in \mathcal{X}\}.$$

- $\mathbb{E}\|R_n\|_{\mathcal{F}} \leq \sqrt{\frac{2\log(2\Pi_{\mathcal{F}}(n))}{n}}$; $\qquad \mathbb{E}\|P - P_n\|_{\mathcal{F}} = O\left(\sqrt{\frac{\log(\Pi_{\mathcal{F}}(n))}{n}}\right)$.

Growth Function

Definition.

For a class $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$, the **growth function** is

$$\Pi_{\mathcal{F}}(n) = \max\{|\mathcal{F}(x_1^n)| : x_1, \ldots, x_n \in \mathcal{X}\}.$$

- $\mathbb{E}\|R_n\|_{\mathcal{F}} \leq \sqrt{\frac{2\log(2\Pi_{\mathcal{F}}(n))}{n}}$; $\mathbb{E}\|P - P_n\|_{\mathcal{F}} = O\left(\sqrt{\frac{\log(\Pi_{\mathcal{F}}(n))}{n}}\right)$.
- $\Pi_{\mathcal{F}}(n) \leq |\mathcal{F}|$, $\lim_{n\to\infty} \Pi_{\mathcal{F}}(n) = |\mathcal{F}|$.

Growth Function

Definition.

For a class $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$, the **growth function** is

$$\Pi_{\mathcal{F}}(n) = \max\{|\mathcal{F}(x_1^n)| : x_1, \dots, x_n \in \mathcal{X}\}.$$

- $\mathbb{E}\|R_n\|_{\mathcal{F}} \leq \sqrt{\frac{2\log(2\Pi_{\mathcal{F}}(n))}{n}}$; $\qquad \mathbb{E}\|P - P_n\|_{\mathcal{F}} = O\left(\sqrt{\frac{\log(\Pi_{\mathcal{F}}(n))}{n}}\right)$.
- $\Pi_{\mathcal{F}}(n) \leq |\mathcal{F}|$, $\lim_{n\to\infty} \Pi_{\mathcal{F}}(n) = |\mathcal{F}|$.
- $\Pi_{\mathcal{F}}(n) \leq 2^n$. (But then this gives no useful bound on $\mathbb{E}\|R_n\|_{\mathcal{F}}$.)

Outline

Vapnik-Chervonenkis Dimension

Definition.

A class $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$ **shatters** $\{x_1, \ldots, x_d\} \subseteq \mathcal{X}$ means that $|\mathcal{F}(x_1^d)| = 2^d$. The Vapnik-Chervonenkis dimension of $\mathcal{F}$ is

$$\begin{aligned} d_{VC}(\mathcal{F}) &= \max\{d : \text{some } x_1, \ldots, x_d \in \mathcal{X} \text{ is shattered by } \mathcal{F}\} \\ &= \max\left\{d : \Pi_{\mathcal{F}}(d) = 2^d\right\}. \end{aligned}$$

Vapnik-Chervonenkis Dimension: "Sauer's Lemma"

Theorem (Vapnik-Chervonenkis)**.**

$d_{VC}(\mathcal{F}) \leq d$ implies

$$\Pi_{\mathcal{F}}(n) \leq \sum_{i=0}^{d} \binom{n}{i}.$$

If $n \geq d$, the latter sum is no more than $\left(\frac{en}{d}\right)^d$.

Vapnik-Chervonenkis Dimension: "Sauer's Lemma"

Theorem (Vapnik-Chervonenkis)**.**

$d_{VC}(\mathcal{F}) \leq d$ implies

$$\Pi_{\mathcal{F}}(n) \leq \sum_{i=0}^{d} \binom{n}{i}.$$

If $n \geq d$, the latter sum is no more than $\left(\frac{en}{d}\right)^d$.

So the VC-dimension d is a single integer summary of the growth function:

$$\Pi_{\mathcal{F}}(n) \begin{cases} = 2^n & \text{if } n \leq d, \\ \leq (e/d)^d \, n^d & \text{if } n > d. \end{cases}$$

Vapnik-Chervonenkis Dimension: “Sauer’s Lemma”

Thus, for $d_{VC}(\mathcal{F}) \leq d$ and $n \geq d$,

$$\sup_P \mathbb{E}\|P - P_n\|_{\mathcal{F}} = O\left(\sqrt{\frac{d\log(n/d)}{n}}\right).$$

And there is a matching lower bound.

Uniform convergence *uniformly over probability distributions* is equivalent to finiteness of the VC-dimension.

VC-dimension Bounds for Parameterized Families

Consider a parameterized class of binary-valued functions,

$$\mathcal{F} = \{x \mapsto f(x, \theta) : \theta \in \mathbb{R}^p\},$$

where $f : \mathcal{X} \times \mathbb{R}^p \to \{\pm 1\}$.

Suppose that for each x, $\theta \mapsto f(x, \theta)$ can be computed using no more than t operations of the following kinds:

1. arithmetic ($+$, $-$, $\times$, $/$),
2. comparisons ($>$, $=$, $<$),
3. output ± 1.

Theorem (Goldberg and Jerrum)**.**

$d_{VC}(\mathcal{F}) \leq 4p(t+2)$.

Outline

VC-Dimension of Neural Networks

Theorem.

Consider the class $\mathcal{F}$ of $\{-1, 1\}$-valued functions computed by a network with L layers, p parameters, and k computation units with the following nonlinearities:

VC-Dimension of Neural Networks

Theorem.

Consider the class $\mathcal{F}$ of $\{-1, 1\}$-valued functions computed by a network with L layers, p parameters, and k computation units with the following nonlinearities:

1. Piecewise constant (linear threshold units):
$$\mathrm{VCdim}(\mathcal{F}) = \tilde{\Theta}(p).$$
(Baum and Haussler, 1989)

VC-Dimension of Neural Networks

Theorem.

Consider the class $\mathcal{F}$ of $\{-1, 1\}$-valued functions computed by a network with L layers, p parameters, and k computation units with the following nonlinearities:

1. Piecewise constant (linear threshold units):
$$\mathrm{VCdim}(\mathcal{F}) = \tilde{\Theta}(p).$$
(Baum and Haussler, 1989)
2. Piecewise linear (ReLUs):
$$\mathrm{VCdim}(\mathcal{F}) = \tilde{\Theta}(pL).$$
(B., Harvey, Liaw, Mehrabian, 2017)

VC-Dimension of Neural Networks

Theorem.

Consider the class $\mathcal{F}$ of $\{-1, 1\}$-valued functions computed by a network with L layers, p parameters, and k computation units with the following nonlinearities:

1. Piecewise constant (linear threshold units):
$$\mathrm{VCdim}(\mathcal{F}) = \tilde{\Theta}(p).$$
(Baum and Haussler, 1989)
2. Piecewise linear (ReLUs): $$\mathrm{VCdim}(\mathcal{F}) = \tilde{\Theta}(pL).$$
(B., Harvey, Liaw, Mehrabian, 2017)
3. Piecewise polynomial: $$\mathrm{VCdim}(\mathcal{F}) = \tilde{O}\left(pL^2\right).$$
(B., Maiorov, Meir, 1998)

VC-Dimension of Neural Networks

Theorem.

Consider the class $\mathcal{F}$ of $\{-1,1\}$-valued functions computed by a network with L layers, p parameters, and k computation units with the following nonlinearities:

1. Piecewise constant (linear threshold units):
$$\mathrm{VCdim}(\mathcal{F}) = \tilde{\Theta}\left(p\right).$$
(Baum and Haussler, 1989)

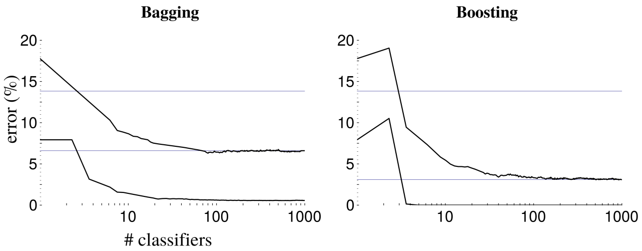
2. Piecewise linear (ReLUs): $$\mathrm{VCdim}(\mathcal{F}) = \tilde{\Theta}\left(pL\right).$$
(B., Harvey, Liaw, Mehrabian, 2017)

3. Piecewise polynomial: $$\mathrm{VCdim}(\mathcal{F}) = \tilde{O}\left(pL^2\right).$$
(B., Maiorov, Meir, 1998)

4. Sigmoid: $$\mathrm{VCdim}(\mathcal{F}) = \tilde{O}\left(p^2k^2\right).$$
(Karpinsky and MacIntyre, 1994)

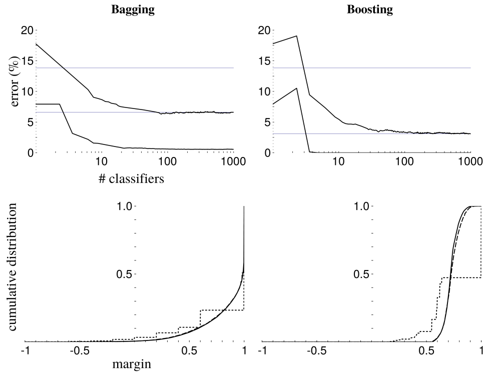
Outline

Some Experimental Observations



(Schapire, Freund, B, Lee, 1997)

Some Experimental Observations



(Schapire, Freund, B, Lee, 1997)

Some Experimental Observations

monly believed to be accurate. However, the stipulation that the number of parameters must be less than the number of examples is typically believed to be true for common datasets. The results here indicate that this is not always the case.

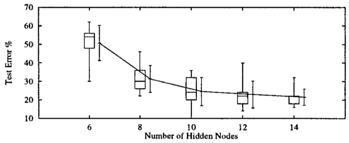
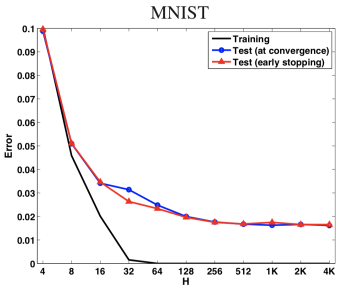
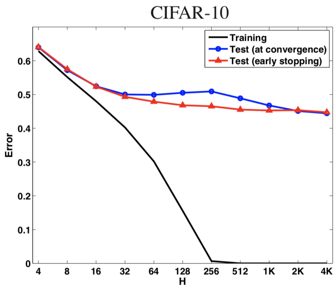


Figure 3. Face recognition example: the best generalizing network has 364 times more parameters than training points (18210 parameters).

(Lawrence, Giles, Tsoi, 1997)

Some Experimental Observations





(Neyshabur, Tomioka, Srebro, 2015)

Large-Margin Classifiers

- Consider a real-valued function $f : \mathcal{X} \to \mathbb{R}$ used for classification.
- The prediction on $x \in \mathcal{X}$ is $\mathrm{sign}(f(x)) \in \{-1, 1\}$.
- If $yf(x) > 0$ then f classifies x correctly.
- We call $yf(x)$ the *margin* of f on x. (c.f. the perceptron)
- We can view a larger margin as a more confident correct classification.
- Minimizing a continuous loss, such as

$$\sum_{i=1}^{n} (f(X_i) - Y_i)^2 , \qquad \text{or} \qquad \sum_{i=1}^{n} \exp(-Y_i f(X_i)) ,$$

  encourages large margins.
- For large-margin classifiers, we should expect the fine-grained details of f to be less important.

Recall

Contraction property:

If ℓ is $\hat{y} \mapsto \ell(\hat{y}, y)$ 1-Lipschitz for all y and $|\ell| \leq 1$, then

$$\mathbb{E}\|R_n\|_{\ell \circ \mathcal{F}} \leq 2\mathbb{E}\|R_n\|_{\mathcal{F}} + c/\sqrt{n}$$

Unfortunately, classification loss is not Lipschitz. By writing classification loss as $(1 - yy')/2$, we did show that

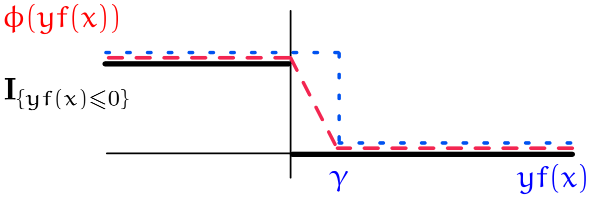
$$\mathbb{E}\|R_n\|_{\ell \circ \mathrm{sign}(\mathcal{F})} \leq 2\mathbb{E}\|R_n\|_{\mathrm{sign}(\mathcal{F})} + c/\sqrt{n}$$

but we want $\mathbb{E}\|R_n\|_{\mathcal{F}}$.

Rademacher Complexity for Lipschitz Loss

Consider the $1/\gamma$-Lipschitz *surrogate* loss

$$\phi(\alpha) = \begin{cases} 1 & \text{if } \alpha \leq 0, \\ 1 - \alpha/\gamma & \text{if } 0 < \alpha < \gamma, \\ 0 & \text{if } \alpha \geq 1. \end{cases}$$



Rademacher Complexity for Lipschitz Loss

Large margin loss is an upper bound and classification loss is a lower bound:

$$\mathbf{I}\{Yf(X) \le 0\} \le \phi(Yf(X)) \le \mathbf{I}\{Yf(X) \le \gamma\}.$$

So if we can relate the Lipschitz risk $P\phi(Yf(X))$ to the Lipschitz empirical risk $P_n\phi(Yf(X))$, we have a large margin bound:

$$P\mathbf{I}\{Yf(X) \le 0\} \le P\phi(Yf(X)) \text{ c.f. } P_n\phi(Yf(X)) \le P_n\mathbf{I}\{Yf(X) \le \gamma\}.$$

Rademacher Complexity for Lipschitz Loss

With high probability, for all $f \in \mathcal{F}$,

$$\begin{aligned} L_{01}(f) &= P\mathbf{1}\{Yf(X) \le 0\} \\ &\le P\phi(Yf(X)) \\ &\le P_n\phi(Yf(X)) + \frac{c}{\gamma}\mathbb{E}\|R_n\|_{\mathcal{F}} + O(1/\sqrt{n}) \\ &\le P_n\mathbf{1}\{Yf(X) \le \gamma\} + \frac{c}{\gamma}\mathbb{E}\|R_n\|_{\mathcal{F}} + O(1/\sqrt{n}). \end{aligned}$$

Notice that we've turned a classification problem into a regression problem.

The VC-dimension (which captures arbitrarily fine-grained properties of the function class) is no longer important.

Example: Linear Classifiers

Let $\mathcal{F} = \{x \mapsto \langle w, x \rangle : \|w\| \le 1\}$. Then, since $\mathbb{E}\,\|R_n\|_{\mathcal{F}} \lesssim n^{-1/2}$,

$$L_{01}(f) \lesssim \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{Y_i f(X_i) \le \gamma\} + \frac{1}{\gamma}\frac{1}{\sqrt{n}}$$

Compare to Perceptron.

Risk versus surrogate risk

In general, these margin bounds are upper bounds only.

But for any *surrogate loss* ϕ, we can define a transform ψ that gives a *tight* relationship between the excess $0-1$ risk (over the Bayes risk) to the excess ϕ-risk:

Theorem.

1. For any P and f, $\quad \psi\left(L_{01}(f) - L_{01}^*\right) \le L_\phi(f) - L_\phi^*$.
2. For $|\mathcal{X}| \ge 2$, $\epsilon > 0$ and $\theta \in [0,1]$, there is a P and an f with

$$\begin{aligned} &L_{01}(f) - L_{01}^* = \theta \\ &\psi(\theta) \le L_\phi(f) - L_\phi^* \le \psi(\theta) + \epsilon. \end{aligned}$$

Generalization: Margins and Size of Parameters

- A *classification* problem becomes a *regression* problem if we use a loss function that doesn't vary too quickly.

Generalization: Margins and Size of Parameters

- A *classification* problem becomes a *regression* problem if we use a loss function that doesn't vary too quickly.
- For regression, the complexity of a neural network is controlled by the *size* of the parameters, and can be independent of the number of parameters.

Generalization: Margins and Size of Parameters

- A *classification* problem becomes a *regression* problem if we use a loss function that doesn't vary too quickly.
- For regression, the complexity of a neural network is controlled by the *size* of the parameters, and can be independent of the number of parameters.
- We have a tradeoff between the fit to the training data (margins) and the complexity (size of parameters):

$$\Pr(\mathrm{sign}(f(X)) \neq Y) \leq \frac{1}{n}\sum_{i=1}^{n} \phi(Y_i f(X_i)) + p_n(f)$$

Generalization: Margins and Size of Parameters

- A *classification* problem becomes a *regression* problem if we use a loss function that doesn't vary too quickly.
- For regression, the complexity of a neural network is controlled by the *size* of the parameters, and can be independent of the number of parameters.
- We have a tradeoff between the fit to the training data (margins) and the complexity (size of parameters):

$$\Pr(\mathrm{sign}(f(X)) \neq Y) \leq \frac{1}{n} \sum_{i=1}^{n} \phi(Y_i f(X_i)) + p_n(f)$$

- Even if the training set is classified correctly, it might be worthwhile to increase the complexity, to improve this loss function.

In the next few sections, we show that

$$\mathbb{E}\left[L(\widehat{f}_n) - \widehat{L}(\widehat{f}_n)\right]$$

can be small if the learning algorithm $\widehat{f}_n$ has certain properties.

Keep in mind that this is an interesting quantity for two reasons:

- it upper bounds the excess expected loss of ERM, as we showed before.
- can be written as an a-posteriori bound

  $$L(\widehat{f}_n) \leq \widehat{L}(\widehat{f}_n) + \dots$$

  regardless of whether $\widehat{f}_n$ minimized empirical loss.

Outline

Compression Set

Let us use the shortened notation for data: $\mathcal{S} = \{Z_1, \ldots, Z_n\}$, and let us make the dependence of the algorithm $\widehat{f}_n$ on the training set explicit: $\widehat{f}_n = \widehat{f}_n[\mathcal{S}]$. As before, denote $\mathcal{G} = \{(x, y) \mapsto \ell(f(x), y) : f \in \mathcal{F}\}$, and write $\widehat{g}_n(\cdot) = \ell(\widehat{f}_n(\cdot), \cdot)$. Let us write $\widehat{g}_n[\mathcal{S}](\cdot)$ to emphasize the dependence.

Suppose there exists a "compression function" C_k which selects from any dataset $\mathcal{S}$ of size n a subset of k examples $C_k(\mathcal{S}) \subseteq \mathcal{S}$ such that

$$\widehat{f}_n[\mathcal{S}] = \widehat{f}_k[C_k(\mathcal{S})]$$

That is, the learning algorithm produces the same function when given $\mathcal{S}$ or its subset $C_k(\mathcal{S})$.

One can keep in mind the example of support vectors in SVMs.

Then,

$$\begin{aligned} L(\widehat{f}_n) - \widehat{L}(\widehat{f}_n) &= \mathbb{E}\widehat{g}_n - \frac{1}{n}\sum_{i=1}^{n} \widehat{g}_n(Z_i) \\ &= \mathbb{E}\widehat{g}_k[C_k(\mathcal{S})](Z) - \frac{1}{n}\sum_{i=1}^{n} \widehat{g}_k[C_k(\mathcal{S})](Z_i) \\ &\le \max_{I \subseteq \{1,\ldots,n\}, |I| \le k} \left\{ \mathbb{E}\widehat{g}_k[\mathcal{S}_I](Z) - \frac{1}{n}\sum_{i=1}^{n} \widehat{g}_k[\mathcal{S}_I](Z_i) \right\} \end{aligned}$$

where $\mathcal{S}_I$ is the subset indexed by I.

Since $\widehat{g}_k[\mathcal{S}_I]$ only depends on k out of n points, the empirical average is "mostly out of sample". Adding and subtracting loss functions for an additional set of i.i.d. random variables $W = \{Z'_1, \ldots, Z'_k\}$ results in an upper bound

$$\max_{I \subseteq \{1,\ldots,n\}, |I| \leq k} \left\{ \mathbb{E}\widehat{g}_k[\mathcal{S}_I](Z) - \frac{1}{n} \sum_{Z' \in \mathcal{S}'} \widehat{g}_k[\mathcal{S}_I](Z') \right\} + \frac{(b-a)k}{n}$$

where $[a, b]$ is the range of functions in $\mathcal{G}$ and $\mathcal{S}'$ is obtained from $\mathcal{S}$ by replacing $\mathcal{S}_I$ with the corresponding subset W_I.

For each fixed I, the random variable

$$\mathbb{E}\widehat{g}_k[\mathcal{S}_I](Z) - \frac{1}{n}\sum_{Z' \in \mathcal{S}'} \widehat{g}_k[\mathcal{S}_I](Z')$$

is zero mean with standard deviation $O((b-a)/\sqrt{n})$. Hence, the expected maximum over I with respect to $\mathcal{S}, W$ is at most

$$c\sqrt{\frac{(b-a)k\log(en/k)}{n}}$$

since $\log\binom{n}{k} \leq k\log(en/k)$.

Conclusion: compression-style argument limits the bias

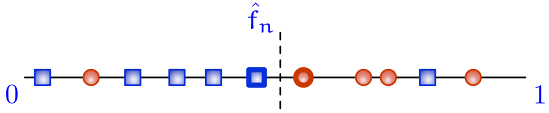
$$\mathbb{E}\left[L(\widehat{f}_n) - \widehat{L}(\widehat{f}_n)\right] \leq O\left(\sqrt{\frac{k \log n}{n}}\right),$$

which is non-vacuous if $k = o(n/\log n)$.

Recall that this term was the upper bound (up to log) on expected excess loss of ERM if class has VC dimension k. However, a possible equivalence between compression and VC dimension is still being investigated.

Example: Classification with Thresholds in 1D

- $\mathcal{X} = [0, 1]$, $\mathcal{Y} = \{0, 1\}$
- $\mathcal{F} = \{f_\theta : f_\theta(x) = \mathbb{I}\{x \geq \theta\}, \theta \in [0, 1]\}$
- $\ell(f_\theta(x), y) = \mathbb{I}\{f_\theta(x) \neq y\}$



For any set of data $(x_1, y_1), \ldots, (x_n, y_n)$, the ERM solution $\widehat{f}_n$ has the property that the first occurrence x_l on the left of the threshold has label $y_l = 0$, while first occurrence x_r on the right – label $y_r = 1$.

Enough to take $k = 2$ and define $\widehat{f}_n[\mathcal{S}] = \widehat{f}_2[(x_l, 0), (x_r, 1)]$.

Further examples/observations:

- Compression of size d for hyperplanes (realizable case)
- Compression of size $1/\gamma^2$ for margin case
- Bernstein bound gives $1/n$ rate rather than $1/\sqrt{n}$ rate on realizable data (zero empirical error).

Outline

Algorithmic Stability

Recall that compression was a way to upper bound $\mathbb{E}\left[L(\widehat{f}_n) - \widehat{L}(\widehat{f}_n)\right]$. Algorithmic stability is another path to the same goal.

Compare:

- Compression: $\widehat{f}_n$ depends only on a subset of k datapoints.
- Stability: $\widehat{f}_n$ does not depend on any of the datapoints too strongly.

As before, let's write shorthand $g = \ell \circ f$ and $\widehat{g}_n = \ell \circ \widehat{f}_n$.

Algorithmic Stability

We now write

$$\mathbb{E}_{\mathcal{S}} L(\widehat{f}_n) = \mathbb{E}_{Z_1,\dots,Z_n,Z}\left\{ \ \widehat{g}_n[Z_1,\dots,Z_n](Z) \ \right\}$$

Again, the meaning of $\widehat{g}_n[Z_1,\dots,Z_n](Z)$: train on $Z_1,\dots,Z_n$ and test on Z.

On the other hand,

$$\begin{aligned}\mathbb{E}_{\mathcal{S}} \widehat{L}(\widehat{f}_n) &= \mathbb{E}_{Z_1,\dots,Z_n}\left\{ \ \frac{1}{n}\sum_{i=1}^{n} \widehat{g}_n[Z_1,\dots,Z_n](Z_i) \ \right\} \\ &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{Z_1,\dots,Z_n}\left\{ \ \widehat{g}_n[Z_1,\dots,Z_n](Z_i) \ \right\} \\ &= \mathbb{E}_{Z_1,\dots,Z_n}\left\{ \ \widehat{g}_n[Z_1,\dots,Z_n](Z_1) \ \right\}\end{aligned}$$

where the last step holds for symmetric algorithms (wrt permutation of training data). Of course, instead of Z_1 we can take any Z_i.

Now comes the renaming trick. It takes a minute to get used to, if you haven't seen it.

Note that $Z_1, \dots, Z_n, Z$ are i.i.d. Hence,

$$\begin{aligned}\mathbb{E}_{\mathcal{S}} L(\widehat{f}_n) &= \mathbb{E}_{Z_1,\dots,Z_n,Z}\left\{ \;\; \widehat{g}_n[Z_1,\dots,Z_n](Z) \;\; \right\} \\ &= \mathbb{E}_{Z_1,\dots,Z_n,Z}\left\{ \;\; \widehat{g}_n[Z, Z_2,\dots,Z_n](Z_1) \;\; \right\}\end{aligned}$$

Therefore,

$$\mathbb{E}\left\{ L(\widehat{f}_n) - \widehat{L}(\widehat{f}_n) \right\} = \mathbb{E}_{Z_1,\dots,Z_n,Z}\left\{ \;\; \widehat{g}_n[Z, Z_2,\dots,Z_n](Z_1) - \widehat{g}_n[Z_1,\dots,Z_n](Z_1) \;\; \right\}$$

Of course, we haven't really done much except re-writing expectation. But the difference

$$\widehat{g}_n[Z, Z_2, \ldots, Z_n](Z_1) - \widehat{g}_n[Z_1, \ldots, Z_n](Z_1)$$

has a "stability" interpretation. If it holds that the output of the algorithm "does not change much" when one datapoint is replaced with another, then the gap $\mathbb{E}\left\{L(\widehat{f}_n) - \widehat{L}(\widehat{f}_n)\right\}$ is small.

Moreover, since everything we've written is an equality, this stability is equivalent to having small gap $\mathbb{E}\left\{L(\widehat{f}_n) - \widehat{L}(\widehat{f}_n)\right\}$.

NB: our aim of ensuring small $\mathbb{E}\left\{L(\widehat{f}_n) - \widehat{L}(\widehat{f}_n)\right\}$ only makes sense if $\widehat{L}(\widehat{f}_n)$ is small (e.g. on average). That is, the analysis only makes sense for those methods that explicitly or implicitly minimize empirical loss (or a regularized variant of it).

It's not enough to be stable. Consider a learning mechanism that ignores the data and outputs $\widehat{f}_n = f_0$, a constant function. Then $\mathbb{E}\left\{L(\widehat{f}_n) - \widehat{L}(\widehat{f}_n)\right\} = 0$ and the algorithm is very stable. However, it does not do anything interesting.

Uniform Stability

Rather than the average notion we just discussed, let's consider a much stronger notion:

We say that algorithm is β uniformly stable if

$$\forall i \in [n], z_1, \ldots, z_n, z', z \quad \left| \widehat{g}_n[\mathcal{S}](z) - \widehat{g}_n[\mathcal{S}^{i,z'}](z) \right| \leq \beta$$

where $\mathcal{S}^{i,z'} = \{z_1, \ldots, z_{i-1}, z', z_{i+1}, \ldots, z_n\}$.

Uniform Stability

Clearly, for any realization of $Z_1, \ldots, Z_n, Z$,

$$\widehat{g}_n[Z, Z_2, \ldots, Z_n](Z_1) - \widehat{g}_n[Z_1, \ldots, Z_n](Z_1) \leq \beta,$$

and so expected loss of a β-uniformly-stable ERM is β-close to its empirical error (in expectation). See recent work by Feldman and Vondrak for sharp high probability statements.

Of course, it is unclear at this point whether a β-uniformly-stable ERM (or near-ERM) exists.

Kernel Ridge Regression

Consider

$$\widehat{f}_n = \operatorname*{argmin}_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2 + \lambda \|f\|_K^2$$

in RKHS $\mathcal{H}$ corresponding to kernel K.

Assume $K(x, x) \leq \kappa^2$ for any x.

Lemma.

Kernel Ridge Regression is β-uniformly stable with $\beta = O\left(\frac{1}{\lambda n}\right)$

Proof (stability of Kernel Ridge Regression)

To prove this, first recall the definition of a σ-strongly convex function ϕ on convex domain $\mathcal{W}$:

$$\forall u, v \in \mathcal{W}, \quad \phi(u) \geq \phi(v) + \langle \nabla\phi(v), u - v \rangle + \frac{\sigma}{2} \|u - v\|^2 .$$

Suppose ϕ, ϕ' are both σ-strongly convex. Suppose w, w' satisfy $\nabla\phi(w) = \nabla\phi'(w') = 0$. Then

$$\phi(w') \geq \phi(w) + \frac{\sigma}{2} \|w - w'\|^2$$

and

$$\phi'(w) \geq \phi'(w') + \frac{\sigma}{2} \|w - w'\|^2$$

As a trivial consequence,

$$\sigma \|w - w'\|^2 \leq [\phi(w') - \phi'(w')] + [\phi'(w) - \phi(w)]$$

Proof (stability of Kernel Ridge Regression)

Now take

$$\phi(f) = \frac{1}{n}\sum_{i\in\mathcal{S}}(f(x_i) - y_i)^2 + \lambda \|f\|_K^2$$

and

$$\phi'(f) = \frac{1}{n}\sum_{i\in\mathcal{S}'}(f(x_i) - y_i)^2 + \lambda \|f\|_K^2$$

where $\mathcal{S}$ and $\mathcal{S}'$ differ in one element: (x_i, y_i) is replaced with (x_i', y_i').

Let $\widehat{f}_n, \widehat{f}_n'$ be the minimizers of ϕ, ϕ', respectively. Then

$$\phi(\widehat{f}_n') - \phi'(\widehat{f}_n') \leq \frac{1}{n}\left((\widehat{f}_n'(x_i) - y_i)^2 - (\widehat{f}_n'(x_i') - y_i')^2\right)$$

and

$$\phi'(\widehat{f}_n) - \phi(\widehat{f}_n) \leq \frac{1}{n}\left((\widehat{f}_n(x_i') - y_i')^2 - (\widehat{f}_n(x_i) - y_i)^2\right)$$

NB: we have been operating with f as vectors. To be precise, one needs to define the notion of strong convexity over $\mathcal{H}$. Let us sweep it under the rug and say that ϕ, ϕ' are 2λ-strongly convex with respect to $\|\cdot\|_K$.

Proof (stability of Kernel Ridge Regression)

Then $\left\|\widehat{f}_n - \widehat{f}'_n\right\|_K^2$ is at most

$$\frac{1}{2\lambda n}\left((\widehat{f}'_n(x_i) - y_i)^2 - (\widehat{f}_n(x_i) - y_i)^2 + (\widehat{f}_n(x'_i) - y'_i)^2 - (\widehat{f}'_n(x'_i) - y'_i)^2\right)$$

which is at most

$$\frac{1}{2\lambda n} C \left\|\widehat{f}_n - \widehat{f}'_n\right\|_\infty$$

where $C = 4(1+c)$ if $|Y_i| \leq 1$ and $|\widehat{f}_n(x_i)| \leq c$.

On the other hand, for any x

$$f(x) = \langle f, K_x\rangle \leq \|f\|_K \|K_x\| = \|f\|_K \sqrt{\langle K_x, K_x\rangle} = \|f\|_K \sqrt{K(x,x)} \leq \kappa \|f\|_K$$

and so

$$\|f\|_\infty \leq \kappa \|f\|_K .$$

Proof (stability of Kernel Ridge Regression)

Putting everything together,

$$\left\|\widehat{f}_n - \widehat{f}_n'\right\|_K^2 \leq \frac{1}{2\lambda n} C \left\|\widehat{f}_n - \widehat{f}_n'\right\|_\infty \leq \frac{\kappa C}{2\lambda n} \left\|\widehat{f}_n - \widehat{f}_n'\right\|_K$$

Hence,

$$\left\|\widehat{f}_n - \widehat{f}_n'\right\|_K \leq \frac{1}{2\lambda n} C \left\|\widehat{f}_n - \widehat{f}_n'\right\|_\infty \leq \frac{\kappa C}{2\lambda n}$$

To finish the claim,

$$(\widehat{f}_n(x_i) - y_i)^2 - (\widehat{f}_n'(x_i) - y_i)^2 \leq C \left\|\widehat{f}_n - \widehat{f}_n'\right\|_\infty \leq \kappa C \left\|\widehat{f}_n - \widehat{f}_n'\right\|_K \leq O\left(\frac{1}{\lambda n}\right)$$

Outline

We consider "local" procedures such as k-Nearest-Neighbors or local smoothing. They have a different bias-variance decomposition (we do not fix a class $\mathcal{F}$).

Analysis will rely on local similarity (e.g. Lipschitz-ness) of regression function f^*.
Idea: to predict y at a given x, look up in the dataset those Y_i for which X_i is "close" to x.

Bias-Variance

It's time to revisit the bias-variance picture. Recall that our goal was to ensure that

$$\mathbb{E}L(\widehat{f}_n) - L(f^*)$$

decreases with data size n, where f^* gives smallest possible L.

For "simple problems" (that is, strong assumptions on P), one can ensure this without the bias-variance decomposition. Examples: Perceptron, linear regression in $d < n$ regime, etc.

However, for more interesting problems, we cannot get this difference to be small without introducing bias, because otherwise the variance (fluctuation of the stochastic part) is too large.

Our approach so far was to split this term into an estimation-approximation error with respect to some class $\mathcal{F}$:

$$\mathbb{E}L(\widehat{f}_n) - L(f_{\mathcal{F}}) + L(f_{\mathcal{F}}) - L(f^*)$$

Bias-Variance

Now we'll study a different bias-variance decomposition, typically used in nonparametric statistics. We will only work with *square loss*.

Rather than fixing $\mathcal{F}$ that controls the estimation error, we fix an algorithm (procedure/estimator) $\widehat{f}_n$ that has some *tunable parameter*.

By definition $\mathbb{E}[Y|X = x] = f^*(x)$. Then we write

$$
\begin{aligned}
\mathbb{E}L(\widehat{f}_n) - L(f^*) &= \mathbb{E}(\widehat{f}_n(X) - Y)^2 - \mathbb{E}(f^*(X) - Y)^2 \\
&= \mathbb{E}(\widehat{f}_n(X) - f^*(X) + f^*(X) - Y)^2 - \mathbb{E}(f^*(X) - Y)^2 \\
&= \mathbb{E}(\widehat{f}_n(X) - f^*(X))^2
\end{aligned}
$$

because the cross term vanishes (check!)

Bias-Variance

Before proceeding, let us discuss the last expression.

$$\begin{aligned}\mathbb{E}(\widehat{f}_n(X) - f^*(X))^2 &= \mathbb{E}_{\mathcal{S}} \int_x (\widehat{f}_n(x) - f^*(x))^2 P(dx) \\ &= \int_x \mathbb{E}_{\mathcal{S}}(\widehat{f}_n(x) - f^*(x))^2 P(dx)\end{aligned}$$

We will often analyze $\mathbb{E}_{\mathcal{S}}(\widehat{f}_n(x) - f^*(x))^2$ for fixed x and then integrate.

The integral is a measure of distance between two functions:

$$\|f - g\|^2_{L_2(P)} = \int_x (f(x) - g(x))^2 P(dx).$$

Bias-Variance

Let us drop $L_2(P)$ from notation for brevity. The bias-variance decomposition can be written as

$$\begin{aligned}\mathbb{E}_{\mathcal{S}} \left\| \widehat{f}_n - f^* \right\|^2_{L_2(P)} &= \mathbb{E}_{\mathcal{S}} \left\| \widehat{f}_n - \mathbb{E}_{Y_{1:n}}[\widehat{f}_n] + \mathbb{E}_{Y_{1:n}}[\widehat{f}_n] - f^* \right\|^2 \\ &= \mathbb{E}_{\mathcal{S}} \left\| \widehat{f}_n - \mathbb{E}_{Y_{1:n}}[\widehat{f}_n] \right\|^2 + \mathbb{E}_{X_{1:n}} \left\| \mathbb{E}_{Y_{1:n}}[\widehat{f}_n] - f^* \right\|^2 ,\end{aligned}$$

because the cross term is zero in expectation.

The first term is variance, the second is squared bias. One typically increases with the parameter, the other decreases.

Parameter is chosen either (a) theoretically or (b) by cross-validation (this is the usual case in practice).

k-Nearest Neighbors

As before, we are given $(X_1, Y_1), \ldots, (X_n, Y_n)$ i.i.d. from P. To make a prediction of Y at a given x, we sort points according to distance $\|X_i - x\|$. Let

$$(X_{(1)}, Y_{(1)}), \ldots, (X_{(n)}, Y_{(n)})$$

be the sorted list (remember this depends on x).

The k-NN estimate is defined as

$$\widehat{f}_n(x) = \frac{1}{k} \sum_{i=1}^{k} Y_{(i)}.$$

Variance: Given x,

$$\widehat{f}_n(x) - \mathbb{E}_{Y_{1:n}}[\widehat{f}_n(x)] = \frac{1}{k}\sum_{i=1}^{k}(Y_{(i)} - f^*(X_{(i)}))$$

which is on the order of $1/\sqrt{k}$. Then variance is of the order $\frac{1}{k}$.

Bias: a bit more complicated. For a given x,

$$\mathbb{E}_{Y_{1:n}}[\widehat{f}_n(x)] - f^*(x) = \frac{1}{k}\sum_{i=1}^{k}(f^*(X_{(i)}) - f^*(x)).$$

Suppose f^* is 1-Lipschitz. Then the square of above is

$$\left(\frac{1}{k}\sum_{i=1}^{k}(f^*(X_{(i)}) - f^*(x))\right)^2 \leq \frac{1}{k}\sum_{i=1}^{k}\left\|X_{(i)} - x\right\|^2$$

So, the bias is governed by how close the closest k random points are to x.

Claim: enough to know the upper bound on the closest point to x among n points.

Argument: for simplicity assume that n/k is an integer. Divide the original (unsorted) dataset into k blocks, n/k size each. Let X^i be the closest point to x in ith block. Then the collection $X^1, \dots, X^k$, a k-subset which is no closer than the set of k nearest neighbors. That is,

$$\frac{1}{k}\sum_{i=1}^{k}\left\|X_{(i)} - x\right\|^2 \leq \frac{1}{k}\sum_{i=1}^{k}\left\|X^i - x\right\|^2$$

Taking expectation (with respect to dataset), the bias term is at most

$$\mathbb{E}\left\{\frac{1}{k}\sum_{i=1}^{k}\left\|X^i - x\right\|^2\right\} = \mathbb{E}\left\|X^1 - x\right\|^2$$

which is expected squared distance from x to the closest point in a random set of n/k points.

If support of X is bounded and $d \geq 3$, then one can estimate

$$\mathbb{E}\left\|X - X_{(1)}\right\|^2 \lesssim n^{-2/d}.$$

That is, we expect the closest neighbor of a random point X to be no further than $n^{-1/d}$ away from one of n randomly drawn points.

Thus, the bias (which is the expected squared distance from x to the closest point in a random set of n/k points) is at most

$$(n/k)^{-2/d}.$$

Putting everything together, the bias-variance decomposition yields

$$\frac{1}{k} + \left(\frac{k}{n}\right)^{2/d}$$

Optimal choice is $k \sim n^{\frac{2}{2+d}}$ and the overall rate of estimation at a given point x is

$$n^{-\frac{2}{2+d}}.$$

Since the result holds for any x, the integrated risk is also

$$\mathbb{E}\left\|\widehat{f}_n - f^*\right\|^2 \lesssim n^{-\frac{2}{2+d}}.$$

Summary

- We sketched the proof that k-Nearest-Neighbors has sample complexity guarantees for prediction or estimation problems with square loss if k is chosen appropriately.
- Analysis is very different from "empirical process" approach for ERM.
- Truly nonparametric!
- No assumptions on underlying density (in $d \geq 3$) beyond compact support. Additional assumptions needed for $d \leq 3$.

Local Kernel Regression: Nadaraya-Watson

Fix a kernel $K : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$. Assume K is zero outside unit Euclidean ball at origin (not true for e^{-x^2}, but close enough).

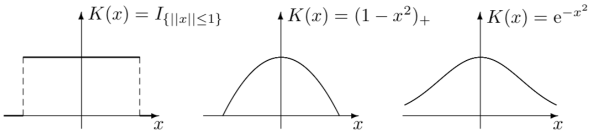


Figure 5.1. Examples for univariate kernels.

(figure from Györfi et al)

Let $K_h(x) = K(x/h)$, and so $K_h(x - x')$ is zero if $\|x - x'\| \geq h$.

h is "bandwidth" – tunable parameter.

Assume $K(x) > c\mathbf{1}\{\|x\| \leq 1\}$ for some $c > 0$. This is important for the "averaging effect" to kick in.

Nadaraya-Watson estimator:

$$\widehat{f}_n(x) = \sum_{i=1}^{n} Y_i W_i(x)$$

with

$$W_i(x) = \frac{K_h(x - X_i)}{\sum_{i=1}^{n} K_h(x - X_i)}$$

(Note: $\sum_i W_i = 1$).

Unlike the k-NN example, bias is easier to estimate.

Bias: for a given x,

$$\mathbb{E}_{Y_{1:n}}[\widehat{f}_n(x)] = \mathbb{E}_{Y_{1:n}}\left[\sum_{i=1}^{n} Y_i W_i(x)\right] = \sum_{i=1}^{n} f^*(X_i) W_i(x)$$

and so

$$\mathbb{E}_{Y_{1:n}}[\widehat{f}_n(x)] - f^*(x) = \sum_{i=1}^{n} (f^*(X_i) - f^*(x)) W_i(x)$$

Suppose f^* is 1-Lipschitz. Since K_h is zero outside the h-radius ball,

$$|\mathbb{E}_{Y_{1:n}}[\widehat{f}_n(x)] - f^*(x)|^2 \leq h^2.$$

Variance: we have

$$\widehat{f}_n(x) - \mathbb{E}_{Y_{1:n}}[\widehat{f}_n(x)] = \sum_{i=1}^{n} (Y_i - f^*(X_i)) W_i(x)$$

Expectation of square of this difference is at most

$$\mathbb{E}\left[\sum_{i=1}^{n} (Y_i - f^*(X_i))^2 W_i(x)^2\right]$$

since cross terms are zero (fix X's, take expectation with respect to the Y's).

We are left analyzing

$$n\mathbb{E}\left[\frac{K_h(x - X_1)^2}{(\sum_{i=1}^{n} K_h(x - X_i))^2}\right]$$

Under some assumptions on density of X, the denominator is at least $(nh^d)^2$ with high prob, whereas $\mathbb{E}K_h(x - X_1)^2 = O(h^d)$ assuming $\int K^2 < \infty$. This gives an overall variance of $O(1/(nh^d))$. *Many* details skipped here (e.g. problems at the boundary, assumptions, etc)

Overall, bias and variance with $h \sim n^{-\frac{1}{2+d}}$ yield

$$h^2 + \frac{1}{nh^d} = n^{-\frac{2}{2+d}}$$

Summary

- Analyzed smoothing methods with kernels. As with nearest neighbors, slow (nonparametric) rates in large d.
- Same bias-variance decomposition approach as k-NN.

Outline

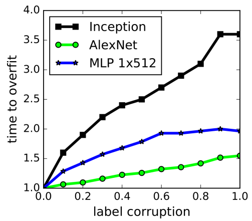
Overfitting in Deep Networks

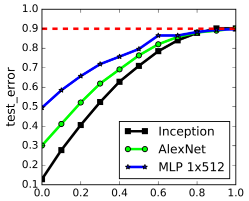
- Deep networks can be trained to zero training error (for *regression* loss)

Overfitting in Deep Networks

- Deep networks can be trained to zero training error (for *regression* loss)
- ... with near state-of-the-art performance (Ruslan Salakhutdinov, Simons Machine Learning Boot Camp, January 2017)

Overfitting in Deep Networks

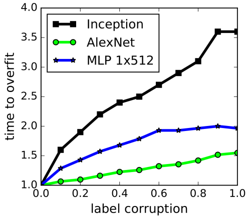


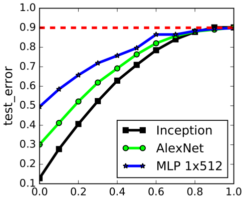


(Zhang, Bengio, Hardt, Recht, Vinyals, 2017)

- Deep networks can be trained to zero training error (for *regression* loss)
- ... with near state-of-the-art performance (Ruslan Salakhutdinov, Simons Machine Learning Boot Camp, January 2017)
- ... even for noisy problems.

Overfitting in Deep Networks

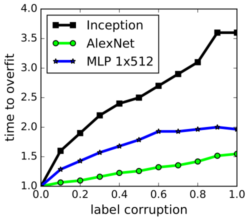


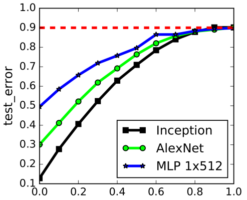


(Zhang, Bengio, Hardt, Recht, Vinyals, 2017)

- Deep networks can be trained to zero training error (for *regression* loss)
- ... with near state-of-the-art performance (Ruslan Salakhutdinov, Simons Machine Learning Boot Camp, January 2017)
- ... even for noisy problems.
- No tradeoff between fit to training data and complexity!

Overfitting in Deep Networks





(Zhang, Bengio, Hardt, Recht, Vinyals, 2017)

- Deep networks can be trained to zero training error (for *regression* loss)
- ... with near state-of-the-art performance (Ruslan Salakhutdinov, Simons Machine Learning Boot Camp, January 2017)
- ... even for noisy problems.
- No tradeoff between fit to training data and complexity!
- *Benign overfitting.*

Overfitting in Deep Networks

Not surprising:

Surprising:

Overfitting in Deep Networks

Not surprising:

- More parameters than sample size. (That's called 'nonparametric.')

Surprising:

Overfitting in Deep Networks

Not surprising:

- More parameters than sample size. (That's called 'nonparametric.')
- Good performance with zero classification loss on training data. (c.f. Margins analysis: trade-off between (regression) fit and complexity.)

Surprising:

Overfitting in Deep Networks

Not surprising:

- More parameters than sample size. (That's called 'nonparametric.')
- Good performance with zero classification loss on training data. (c.f. Margins analysis: trade-off between (regression) fit and complexity.)
- Good performance with zero regression loss on training data when the Bayes risk is zero. (c.f. Perceptron)

Surprising:

Overfitting in Deep Networks

Not surprising:

- More parameters than sample size. (That's called 'nonparametric.')
- Good performance with zero classification loss on training data. (c.f. Margins analysis: trade-off between (regression) fit and complexity.)
- Good performance with zero regression loss on training data when the Bayes risk is zero. (c.f. Perceptron)

Surprising:

- Good performance with zero regression loss on *noisy* training data.

Overfitting in Deep Networks

Not surprising:

- More parameters than sample size. (That's called 'nonparametric.')
- Good performance with zero classification loss on training data. (c.f. Margins analysis: trade-off between (regression) fit and complexity.)
- Good performance with zero regression loss on training data when the Bayes risk is zero. (c.f. Perceptron)

Surprising:

- Good performance with zero regression loss on *noisy* training data.
- All the label noise is making its way into the parameters, but it isn't hurting prediction accuracy!

Overfitting

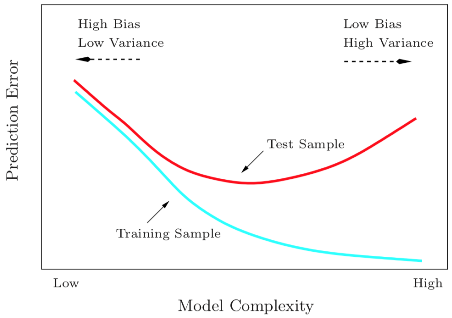


FIGURE 2.11. *Test and training error as a function of model complexity.*

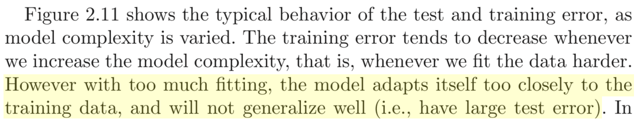
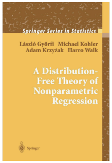


Figure 2.11 shows the typical behavior of the test and training error, as model complexity is varied. The training error tends to decrease whenever we increase the model complexity, that is, whenever we fit the data harder. However with too much fitting, the model adapts itself too closely to the training data, and will not generalize well (i.e., have large test error). In

"... interpolating fits... [are] unlikely to predict future data well at all."

"Elements of Statistical Learning," Hastie, Tibshirani, Friedman, 2001

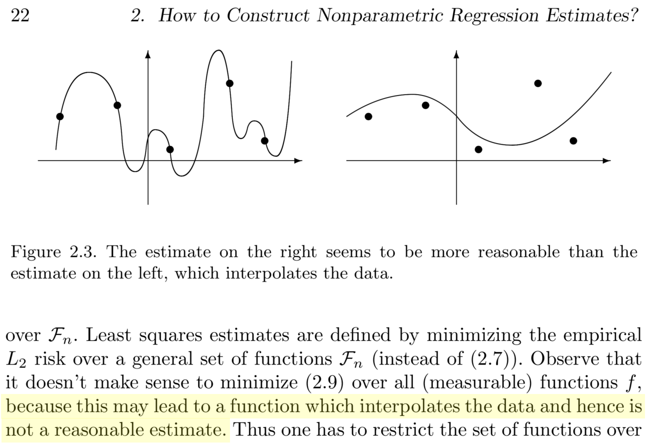
Overfitting


Springer Series in Statistics

László Györfi Michael Kohler
Adam Krzyżak Harro Walk

A Distribution-Free Theory of Nonparametric Regression

Springer



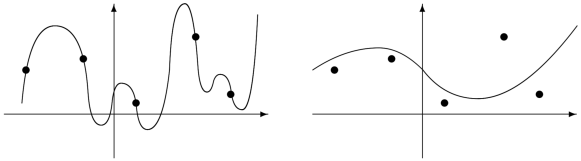


Figure 2.3. The estimate on the right seems to be more reasonable than the estimate on the left, which interpolates the data.

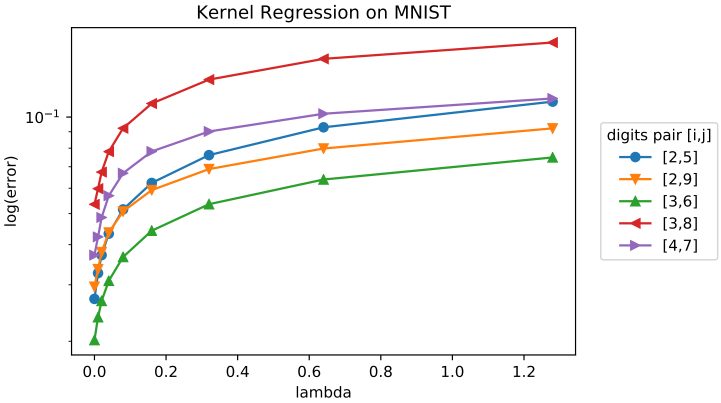
over $\mathcal{F}_n$. Least squares estimates are defined by minimizing the empirical L_2 risk over a general set of functions $\mathcal{F}_n$ (instead of (2.7)). Observe that it doesn't make sense to minimize (2.9) over all (measurable) functions f, because this may lead to a function which interpolates the data and hence is not a reasonable estimate. Thus one has to restrict the set of functions over

Benign Overfitting

A new statistical phenomenon:
good prediction with zero training error for regression loss

- Statistical wisdom says a prediction rule should not fit too well.
- But deep networks are trained to fit noisy data perfectly, and they predict well.

Not an Isolated Phenomenon

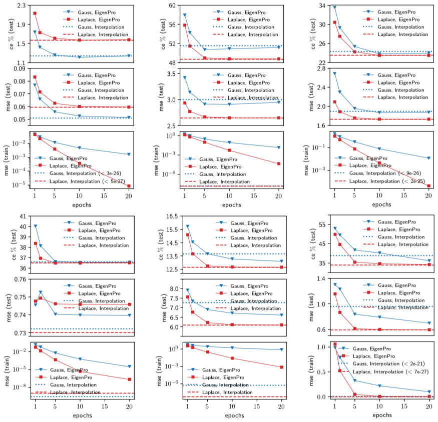


$\lambda = 0$: the interpolated solution, perfect fit on training data.

Not an Isolated Phenomenon

To Understand Deep Learning We Need to Understand
Kernel Learning

Mikhail Belkin, Siyuan Ma, Soumik Mandal
Department of Computer Science and Engineering
Ohio State University
{mbelkin, masi}@cse.ohio-state.edu, mandal.32@osu.edu



(d) TIMIT ($5 \cdot 10^4$ subsamples) (e) HINT-S ($2 \cdot 10^4$ subsamples) (f) 20 Newsgroups

Outline

Simplicial interpolation (Belkin-Hsu-Mitra)

Observe: 1-Nearest Neighbor is interpolating. However, one cannot guarantee $\mathbb{E}L(\widehat{f}_n) - L(f^*)$ small.

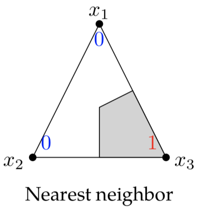
Cover-Hart '67:

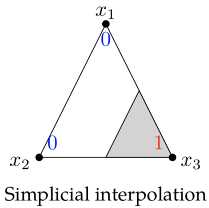
$$\lim_{n\to\infty} \mathbb{E}L(\widehat{f}_n) \leq 2L(f^*)$$

Simplicial interpolation (Belkin-Hsu-Mitra)

Under regularity conditions, simplicial interpolation $\widehat{f}_n$

$$\limsup_{n\to\infty} \mathbb{E} \left\| \widehat{f}_n - f_* \right\|^2 \le \frac{2}{d+2} \mathbb{E}(f^*(X) - Y)^2$$





Blessing of high dimensionality!

Outline

Nadaraya-Watson estimator (Belkin-R.-Tsybakov)

Consider the Nadaraya-Watson estimator. Take a kernel that approaches a large value τ at 0, e.g.

$$K(x) = \min\{1/\|x\|^{\alpha}, \tau\}$$

Note that large τ means $\widehat{f}_n(X_i) \approx Y_i$ since the weight $W_i(X_i)$ is large.

If $\tau = \infty$, we get *interpolation* $\widehat{f}_n(X_i) = Y_i$ of all training data. Yet, earlier proof still goes through. Hence, "memorizing the data" (governed by parameter τ) is completely decoupled from the bias-variance trade-off (as given by parameter h).

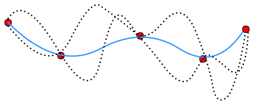
Minimax rates are possible (with a suitable choice of h).

Outline

Min-norm interpolation in RKHS

Min-norm interpolation:

$$\widehat{f}_n := \operatorname*{argmin}_{f \in \mathcal{H}} \|f\|_{\mathcal{H}}, \quad \text{s.t.} \quad f(x_i) = y_i, \ \forall i \le n \ .$$



Closed-form solution:

$$\widehat{f}_n(x) = K(x, X) K(X, X)^{-1} Y$$

Note: GD/SGD started from 0 converges to this minimum-norm solution.

Difficulty in analyzing bias/variance of $\widehat{f}_n$: it is not clear how the random matrix K^{-1} "aggregates" Y's between data points.

Min-norm interpolation in $\mathbb{R}^d$ with $d \asymp n$

(Hastie, Montanari, Rosset, Tibshirani, 2019)

Linear regression with $p, n \to \infty$, $p/n \to \gamma$, empirical spectral distribution of Σ (the discrete measure on its set of eigenvalues) converges to a fixed measure.
Apply random matrix theory to explore the asymptotics of the excess prediction error as γ, the noise variance, and the eigenvalue distribution vary.
Also study the asymptotics of a model involving random nonlinear features.

Implicit regularization in regime $d \asymp n$

(Liang and R., 2018)

Informal Theorem: **Geometric properties of the data design X, high dimensionality**, and **curvature of the kernel** $\Rightarrow$ interpolated solution generalizes.

Bias bounded in terms of effective dimension

$$\dim(XX^\top) = \sum_j \frac{\lambda_j\left(\frac{XX^\top}{d}\right)}{\left[\gamma + \lambda_j\left(\frac{XX^\top}{d}\right)\right]^2}$$

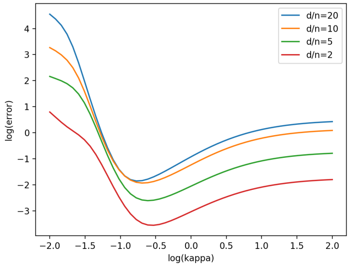
where $\gamma > 0$ is due to implicit regularization due to curvature of kernel and high dimensionality.

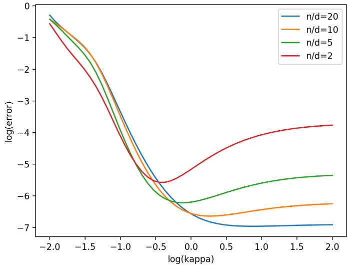
Compare with regularized least squares as low pass filter:

$$\dim(XX^\top) = \sum_j \frac{\lambda_j\left(\frac{XX^\top}{d}\right)}{\lambda + \lambda_j\left(\frac{XX^\top}{d}\right)}$$

Implicit regularization in regime $d \asymp n$

Test error (y-axis) as a function of how quickly spectrum decays. Best performance if spectrum decays but not too quickly.





$$\lambda_j(\Sigma) = \left(1 - \left(\frac{j}{d}\right)^{\kappa}\right)^{1/\kappa}$$

Lower Bounds

(R. and Zhai, 2018)

Min-norm solution for Laplace kernel

$$K_\sigma(x,x') = \sigma^{-d}\exp\{-\|x-x'\|/\sigma\}$$

is not consistent if d is low:

Theorem.

Theorem: for odd dimension d, with probability $1-O(n^{-1/2})$, for any choice of σ,

$$\mathbb{E}\left\|\widehat{f}_n - f^*\right\|^2 \geq \Omega_d(1).$$

Conclusion: need high dimensionality of data.

Connection to neural networks

For wide enough randomly initialized neural networks, GD dynamics quickly converge to (approximately) *min-norm interpolating solution* with respect to a certain kernel.

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} a_i \sigma(\langle w_i, x \rangle)$$

For square loss and relu σ,

$$\frac{\partial \widehat{L}}{\partial w_i} = \frac{1}{\sqrt{m}} \frac{1}{n} \sum_{j=1}^{n} (f(x_j) - y_j) a_i \sigma'(\langle w_i, x_j \rangle) x_j$$

GD in continuous time:

$$\frac{d}{dt} w_i(t) = -\frac{\partial \widehat{L}}{\partial w_i}$$

Thus

$$\frac{d}{dt} f(x) = \sum_{i=1}^{m} \left(\frac{\partial f(x)}{\partial w_i} \right)^T \left(\frac{dw_i}{d_t} \right)$$

Connection to neural networks

For wide enough randomly initialized neural networks, GD dynamics quickly converge to (approximately) *min-norm interpolating solution* with respect to a certain kernel.

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} a_i \sigma(\langle w_i, x \rangle)$$

For square loss and relu σ,

$$\frac{\partial \widehat{L}}{\partial w_i} = \frac{1}{\sqrt{m}} \frac{1}{n} \sum_{j=1}^{n} (f(x_j) - y_j) a_i \sigma'(\langle w_i, x_j \rangle) x_j$$

GD in continuous time:

$$\frac{d}{dt} w_i(t) = -\frac{\partial \widehat{L}}{\partial w_i}$$

Thus

$$\frac{d}{dt} f(x) = \sum_{i=1}^{m} \left(\frac{1}{\sqrt{m}} a_i \sigma'(\langle w_i, x \rangle) x \right)^T \left(-\frac{1}{\sqrt{m}} \frac{1}{n} \sum_{j=1}^{n} (f(x_j) - y_j) a_i \sigma'(\langle w_i, x_j \rangle) x_j \right)$$

Connection to neural networks

For wide enough randomly initialized neural networks, GD dynamics quickly converge to (approximately) *min-norm interpolating solution* with respect to a certain kernel.

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} a_i \sigma(\langle w_i, x \rangle)$$

For square loss and relu σ,

$$\frac{\partial \widehat{L}}{\partial w_i} = \frac{1}{\sqrt{m}} \frac{1}{n} \sum_{j=1}^{n} (f(x_j) - y_j) a_i \sigma'(\langle w_i, x_j \rangle) x_j$$

GD in continuous time:

$$\frac{d}{dt} w_i(t) = -\frac{\partial \widehat{L}}{\partial w_i}$$

Thus

$$\frac{d}{dt} f(x) = -\frac{1}{n} \sum_{j=1}^{n} (f(x_j) - y_j) \left[\frac{1}{m} \sum_{i=1}^{m} a_i^2 \sigma'(\langle w_i, x \rangle) \sigma'(\langle w_i, x_j \rangle) \langle x, x_j \rangle \right]$$

Connection to neural networks

For wide enough randomly initialized neural networks, GD dynamics quickly converge to (approximately) *min-norm interpolating solution* with respect to a certain kernel.

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} a_i \sigma(\langle w_i, x \rangle)$$

For square loss and relu σ,

$$\frac{\partial \widehat{L}}{\partial w_i} = \frac{1}{\sqrt{m}} \frac{1}{n} \sum_{j=1}^{n} (f(x_j) - y_j) a_i \sigma'(\langle w_i, x_j \rangle) x_j$$

GD in continuous time:

$$\frac{d}{dt} w_i(t) = -\frac{\partial \widehat{L}}{\partial w_i}$$

Thus

$$\frac{d}{dt} f(x) = -\frac{1}{n} \sum_{j=1}^{n} (f(x_j) - y_j) \underbrace{K^m(x, x_j)}_{\to K^\infty(x, x_j)}$$

Connection to neural networks

(Xie, Liang, Song, '16), (Jacot, Gabriel, Hongler '18), (Du, Zhai, Poczos, Singh '18), etc.

$$K^{\infty}(x, x_j) = \mathbb{E}[\langle x, x_j \rangle \, \mathbf{I}\{\langle w, x \rangle \geq 0, \langle w, x_j \rangle \geq 0\}]$$

See Jason's talk.

Linear Regression

(B., Long, Lugosi, Tsigler, 2019)

- (x, y) Gaussian, mean zero.
- Define:

$$\Sigma := \mathbb{E}xx^\top = \sum_i \lambda_i v_i v_i^\top, \qquad \text{(assume } \lambda_1 \geq \lambda_2 \geq \cdots\text{)}$$

$$\theta^* := \arg\min_\theta \mathbb{E}\left(y - x^\top\theta\right)^2,$$

$$\sigma^2 := \mathbb{E}(y - x^\top\theta^*)^2.$$

- Estimator $\hat{\theta} = \left(X^\top X\right)^\dagger X^\top y$, which solves

$$\begin{aligned} &\min_\theta && \|\theta\|^2 \\ &\text{s.t.} && \|X\theta - y\|^2 = \min_\beta \|X\beta - y\|^2 \end{aligned}$$

Linear Regression

(B., Long, Lugosi, Tsigler, 2019)

- (x, y) Gaussian, mean zero.
- Define:

$$\Sigma := \mathbb{E}xx^\top = \sum_i \lambda_i v_i v_i^\top, \qquad \text{(assume } \lambda_1 \geq \lambda_2 \geq \cdots \text{)}$$

$$\theta^* := \arg\min_\theta \mathbb{E}\left(y - x^\top\theta\right)^2,$$

$$\sigma^2 := \mathbb{E}(y - x^\top\theta^*)^2.$$

- Estimator $\hat{\theta} = \left(X^\top X\right)^\dagger X^\top y$, which solves

$$\begin{aligned} \min_\theta \quad & \|\theta\|^2 \\ \text{s.t.} \quad & \|X\theta - y\|^2 = \min_\beta \|X\beta - y\|^2 = 0. \end{aligned}$$

- When can the label noise be hidden in $\hat{\theta}$ without hurting predictive accuracy?

Linear Regression

Excess prediction error:

$$L(\hat{\theta}) - L^* := \mathbb{E}_{(x,y)}\left[\left(y - x^\top\hat{\theta}\right)^2 - \left(y - x^\top\theta^*\right)^2\right] = \left(\hat{\theta} - \theta^*\right)^\top \Sigma \left(\hat{\theta} - \theta^*\right).$$

Σ determines how estimation error in different parameter directions affect prediction accuracy.

Linear Regression

Theorem.

For universal constants b, c, and any θ^*, σ^2, Σ, if $\lambda_n > 0$ then

$$\frac{\sigma^2}{c} \min\left\{\frac{k^*}{n} + \frac{n}{R_{k^*}(\Sigma)}, 1\right\}$$
$$\leq \mathbb{E}L(\hat{\theta}) - L^* \leq c\left(\|\theta^*\|^2\sqrt{\frac{\operatorname{tr}(\Sigma)}{n}} + \sigma^2\left(\frac{k^*}{n} + \frac{n}{R_{k^*}(\Sigma)}\right)\right),$$

where $k^* = \min\{k \geq 0 : r_k(\Sigma) \geq bn\}$.

Notions of Effective Rank

Definition.

Recall that $\lambda_1 \geq \lambda_2 \geq \cdots$ are the eigenvalues of Σ. Define the effective ranks

$$r_k(\Sigma) = \frac{\sum_{i>k} \lambda_i}{\lambda_{k+1}}, \qquad R_k(\Sigma) = \frac{\left(\sum_{i>k} \lambda_i\right)^2}{\sum_{i>k} \lambda_i^2}.$$

Notions of Effective Rank

Definition.

Recall that $\lambda_1 \geq \lambda_2 \geq \cdots$ are the eigenvalues of Σ. Define the effective ranks

$$r_k(\Sigma) = \frac{\sum_{i>k} \lambda_i}{\lambda_{k+1}}, \qquad R_k(\Sigma) = \frac{\left(\sum_{i>k} \lambda_i\right)^2}{\sum_{i>k} \lambda_i^2}.$$

Example

If $\mathrm{rank}(\Sigma) = p$, we can write

$$r_0(\Sigma) = \mathrm{rank}(\Sigma) s(\Sigma), \qquad R_0(\Sigma) = \mathrm{rank}(\Sigma) S(\Sigma),$$

$$\text{with} \qquad s(\Sigma) = \frac{1/p \sum_{i=1}^{p} \lambda_i}{\lambda_1}, \qquad S(\Sigma) = \frac{\left(1/p \sum_{i=1}^{p} \lambda_i\right)^2}{1/p \sum_{i=1}^{p} \lambda_i^2}.$$

Both s and S lie between $1/p$ ($\lambda_2 \approx 0$) and 1 (λ_i all equal).

Benign Overfitting in Linear Regression

Intuition

- To avoid harming prediction accuracy, the noise energy must be distributed across many unimportant directions:
  - There should be many non-zero eigenvalues.
  - They should have a small sum compared to n.
  - There should be many eigenvalues no larger than λ_{k^*} (the number depends on how assymmetric they are).

Overfitting?

- Fitting data “too” well versus
- Bias too low, variance too high.

Key takeaway: we should not conflate these two.

Outline