高等机器学习

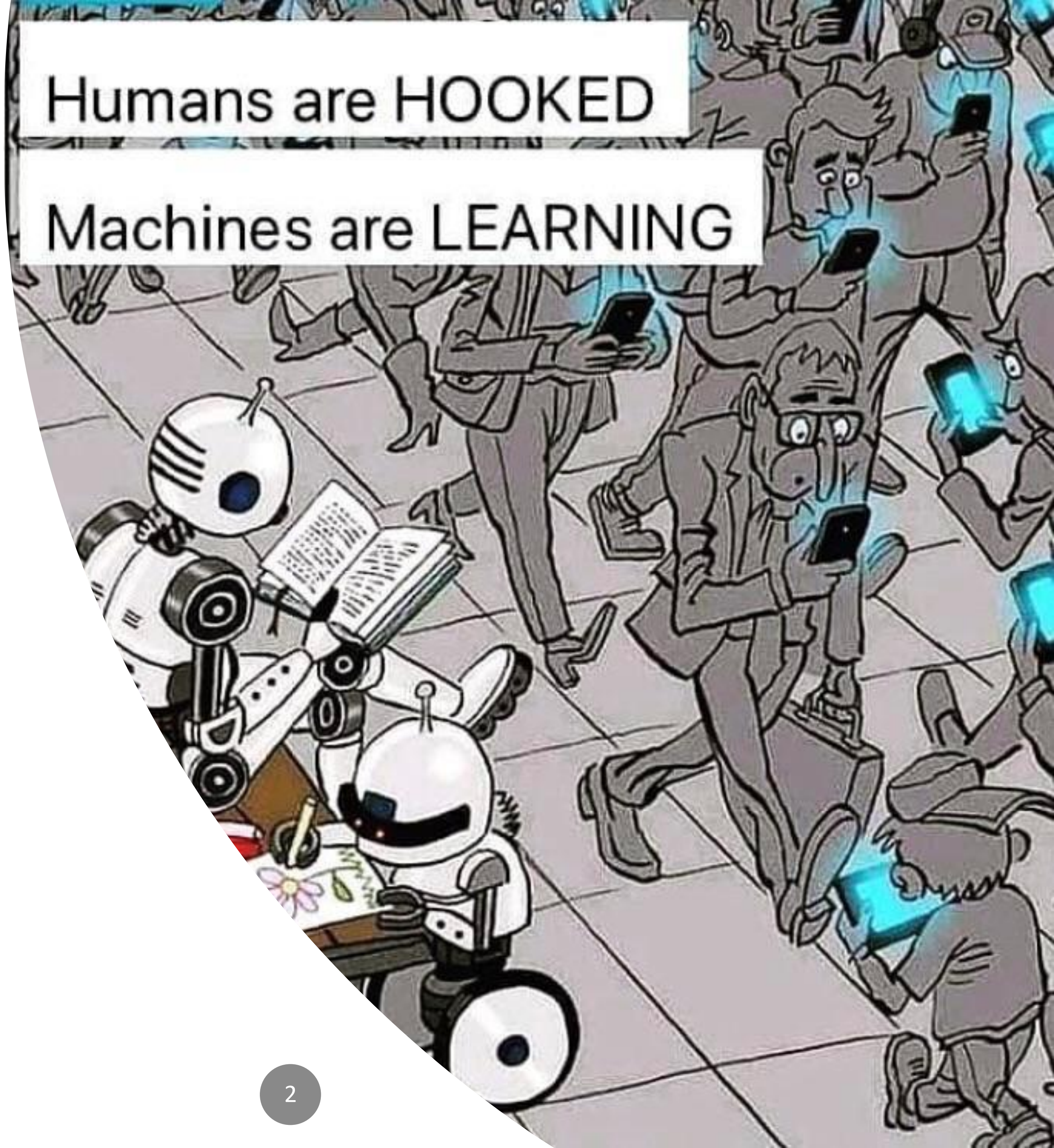
机器学习框架

刘铁岩、张辉帅
微软亚洲研究院

Microsoft

清華大學
Tsinghua University

Statistical Learning Theory

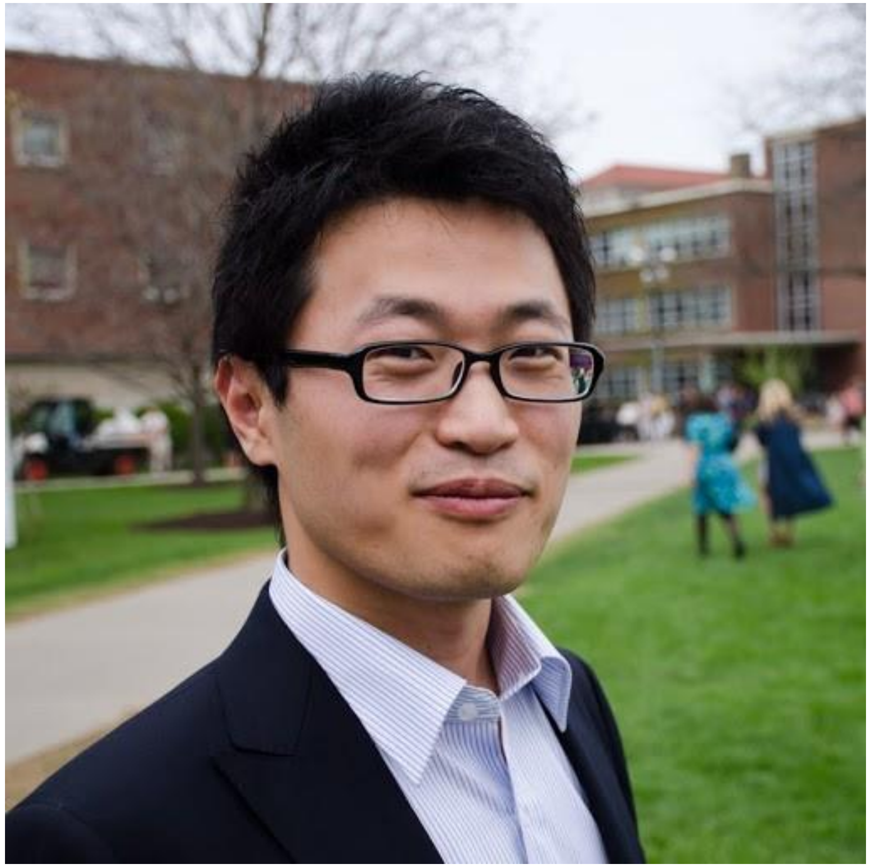


Huishuai Zhang (张辉帅)

Role / Group:	Senior researcher at MSRA
University:	Syracuse University, USTC
Major / Degree:	ECE/PhD
Specialization:	Machine learning theory, optimization

A sentence that best describes you:

A researcher tends to be productive…



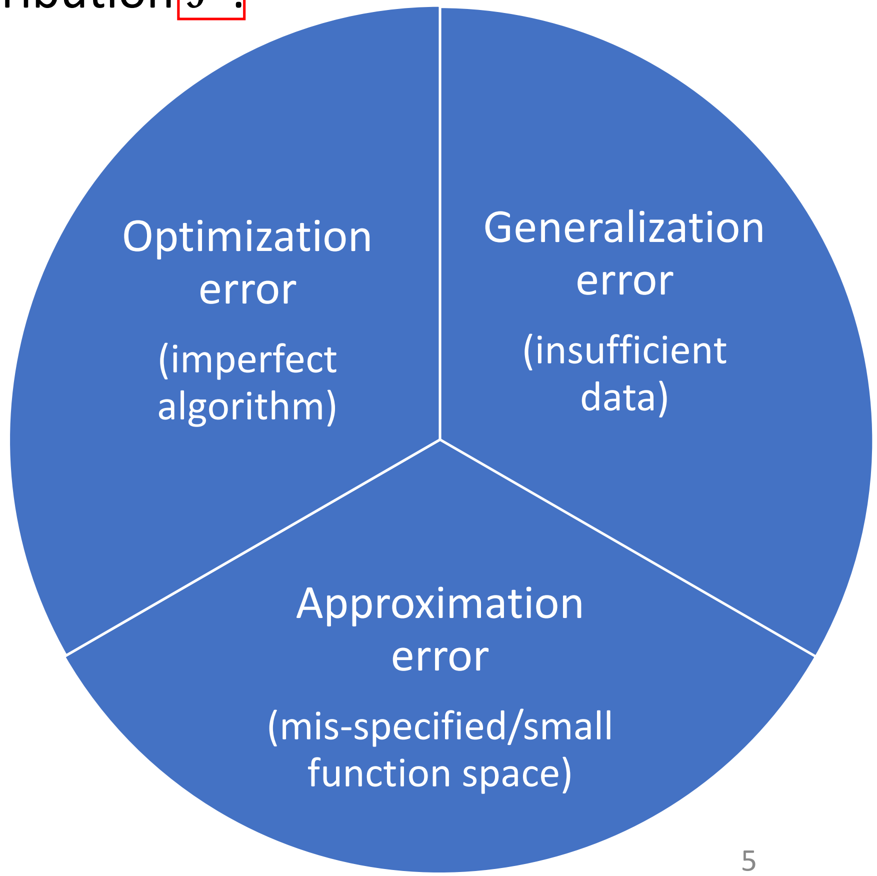
Overall Picture of SLT

- Goal: Good performance on the test data
- Mind the gap
- Bound the gap

Overall Picture of SLT

- Training: Find a function f from a function class $\mathcal{F}$ based on training dataset $\mathcal{D}$.
- Evaluation: How does f perform on test data from the true distribution $\mathcal{P}$?

- Where is the gap?
  - $\mathcal{D} \to \mathcal{P}$: generalization error
  - **Hypothesis space $\mathcal{F}$**: approximation error
  - **Find**: optimization error



Empirical Risk Minimization

- Training data: $\mathrm{D_n} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n$, generated from ground truth distribution P.
- Model: $f \in \mathcal{F}: \mathcal{X} \to \mathcal{Y}$
- *Loss function*: $l(f; x, y) \triangleq l(f(x), y)$
- *(Expected) Risk*: $L(f) = \mathbb{E}_{x,y\sim P} L(f(x), y)$
- *Empirical risk* and *Empirical risk minimization*:

$$\widehat{L_n}(f) = \frac{1}{n}\sum_{i=1}^{n} l(f; x_i, y_i)$$

$$f_n^* = \arg\min_{f\in\mathcal{F}} \widehat{L_n}(f),$$

$$f_n^T \textit{ is the model produced by the learning algorithm at the } T's \textit{ iteration}.$$

- Measure: the *excess risk* of the learnt model f_n^T is defined as

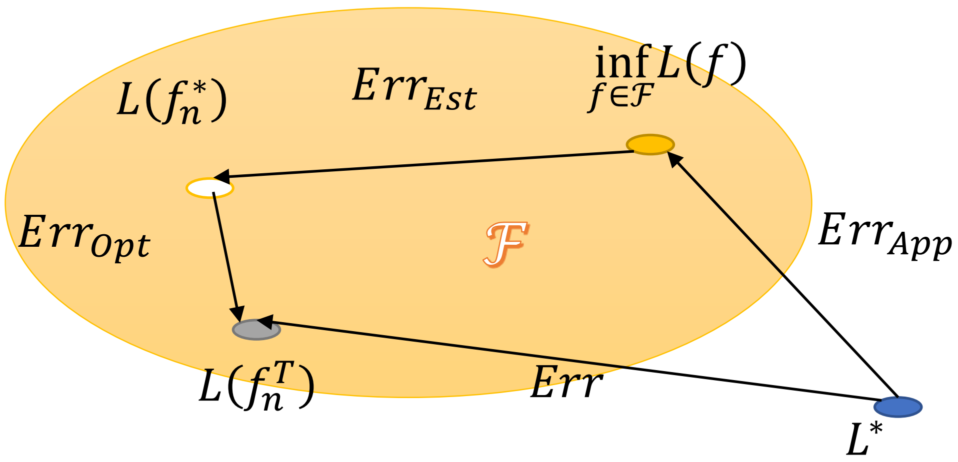
$$L(f_n^T) - L^*$$

where $L^* = \min_f L(f)$ is the Bayes (expected) risk. We want the excess risk as small as possible.

Error Decomposition

Excess risk: Optimization Error, Estimation Error, Approximation Error

$$L(f_n^T) - L^* = \boxed{\left(L(f_n^T) - L(f_n^*)\right)} + \boxed{\left(L(f_n^*) - \inf_{f\in\mathcal{F}} L(f)\right)} + \boxed{\left(\inf_{f\in\mathcal{F}} L(f) - L^*\right)}$$



Discussion

	Optimization Error	Estimation Error	Approximation Error
Definition	$L(f_n^T) - L(f_n^*)$	$L(f_n^*) - \inf_{f \in \mathcal{F}} L(f)$	$\inf_{f \in \mathcal{F}} L(f) - L^*$
Caused by	Approximate Optimization Algorithm	Finite Training Data	Limited Hypothesis Space
Hypothesis space $\mathcal{F}$	Not clear	the larger, the larger	the larger, the smaller
Number of training instances n	In general, the larger, the smaller, but with larger computation cost.	the larger, the smaller	
Opt Algorithm and Iteration number T	the better/larger, the smaller		

Bias and Variance Tradeoff

Optimization and Generalization Interplay

Guarantees for Three Errors

- Optimization error <= = Convergence rate of optimization algorithms
$$L(f_n^T) - L(f_n^*) \leq \epsilon(Alg, \mathcal{F}, n, T)$$
- Estimation/generalization error <= = Upper bound in terms of capacity
$$L(f_n^*) - \inf_{f \in \mathcal{F}} L(f) \leq 2 \sup_{f \in \mathcal{F}} \left| \widehat{L_n}(f) - L(f) \right| \leq \epsilon(Cap(\mathcal{F}), n)$$
- Approximation error (cannot be controllable in general) for neural networks <== Universal approximation theorem of neural networks

Outline

- Optimization theory
- Generalization theory
- Approximation theory

Definition of Convergence Rate

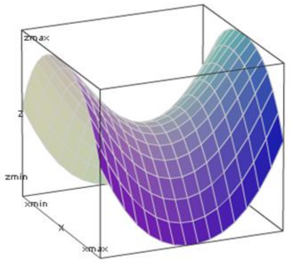
Assume the optimization error $L(f_n^T) - L(f_n^*) \leq \epsilon(Alg, \mathcal{F}, n, T)$

Does the log error $\log \epsilon(T)$ *decrease faster than* $-T$*?*

- Equal to: linear convergence rate, e.g., $O(e^{-T})$
- Faster than: super-linear convergence rate, e.g., $O\left(e^{-T^2}\right)$
  - Quadratic: $\log\log \epsilon(T)$ deceasing in the same order with $-T$, e.g. $O\left(e^{-2^T}\right)$
- Slower than: sub-linear convergence rate, e.g., $O\left(\frac{1}{T}\right)$

Convexity

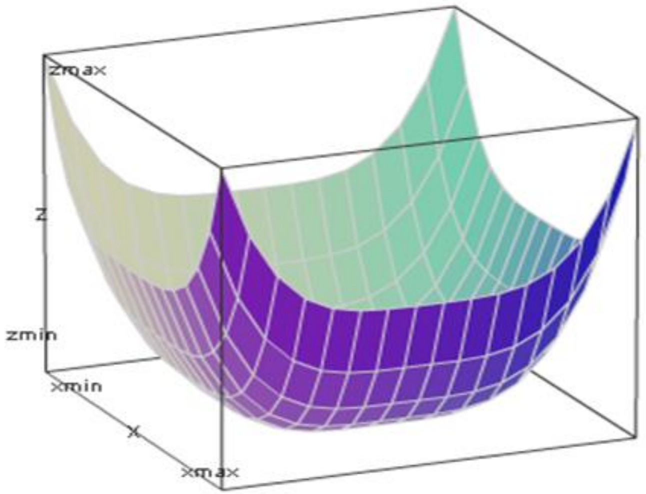
$$g(w) = w_1^2 - w_2^2$$

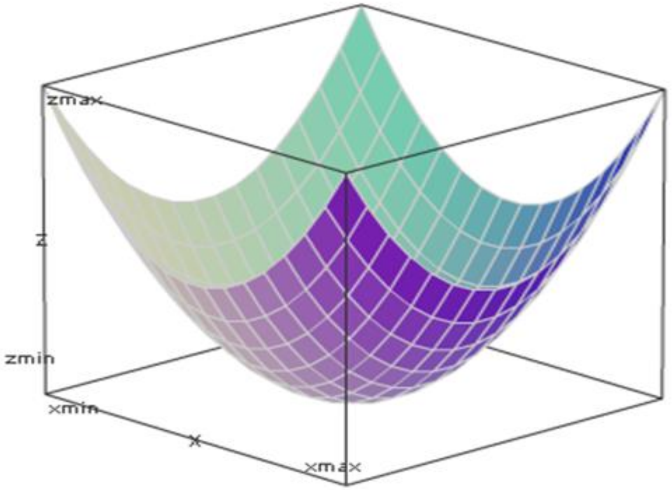


$$g(w) - g(v) \geq \nabla g(v)^\tau (w - v)$$
$$\forall w, v \in \mathcal{W},$$

$$g(w) - g(v) \geq \nabla g(v)^\tau (w - v) + \frac{\alpha}{2} ||w - v||^2$$
$$\forall w, v \in \mathcal{W},$$

$$g(w) = w_1^4 + w_2^4$$





$$g(w) = w_1^2 + w_2^2$$

Convex

Strongly-Convex

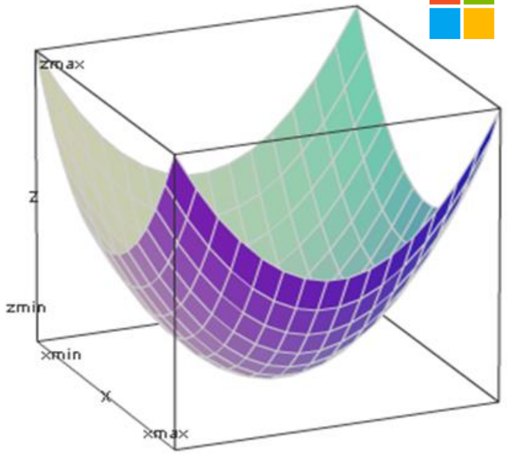
Smoothness

Smooth

β-smooth: $\left|\left|\nabla g(w) - \nabla g(w)\right|\right| \leq \beta\left|\left|w - v\right|\right|$
$\forall w, v \in \mathcal{W},$

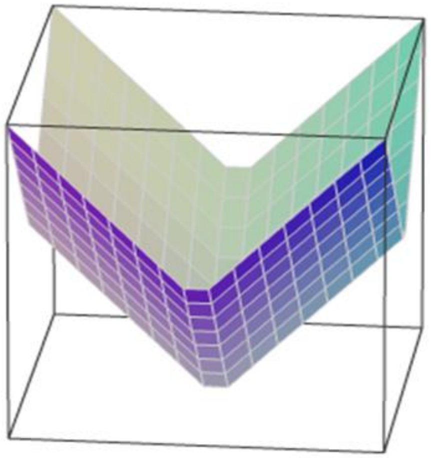
$$g(w) - g(v) \leq \nabla g(v)^{\tau}(w - v) + \frac{\beta}{2}\left|\left|w - v\right|\right|^2$$
$$g(w) - g(v) \geq \nabla g(v)^{\tau}(w - v) + \frac{\alpha}{2}\left|\left|w - v\right|\right|^2$$

For $w^* \in \arg\min f(x)$, we have
$$\frac{\alpha}{2}||w - w^*||^2 \leq g(w) - g(w^*) \leq \frac{\beta}{2}||w - w^*||^2$$
$$(\alpha \leq \beta)$$



Lipschitz

L-Lipschitz: $|g(w) - g(v)| \leq L\left|\left|w - v\right|\right|$
$\forall w, v \in \mathcal{W}$



Convergence Rate of GD

Theorem 1: Assume the objective f is **convex** and β-smooth on R^d.
With step size $\eta = \frac{1}{\beta}$, Gradient Descent satisfies:

$$f(x_{T+1}) - f(x^*) \le \frac{2\beta||x_1 - x^*||^2}{T}.$$

Sub-linear Convergence

Theorem 2: Assume the objective f is **α-strongly convex** and β-smooth on R^d.
With step size $\eta = \frac{2}{\alpha+\beta}$, Gradient Descent satisfies:

$$f(x_{T+1}) - f(x^*) \le \frac{\beta}{2} exp\left(-\frac{4T}{Q+1}\right)||x_1 - x^*||^2,$$

where $Q = \frac{\beta}{\alpha}$.

Linear Convergence

Convergence Rate of Newton's Method

Theorem 3: Suppose the function f is continuously differentiable, its derivative is not 0 at its optimum x^*, and it has a second derivative at x^*, then the convergence is quadratic:

$$||x_t - x^*|| \leq O\left(e^{-2^T}\right)$$

Advantage:

We have a more accurate local approximation of the objective, the convergence is much faster.

Disadvantage:

We need to compute the inverse of Hessian, which is time/storage consuming.

Convergence Rate of GD and SGD

Overall Complexity (ϵ) = Convergence Rate$^{-1}(\epsilon) *$ Complexity of each iteration

	Strongly Convex + Smooth			**Convex + Smooth**		
	Convergence Rate	Complexity of each iteration	Overall Complexity	Convergence Rate	Complexity of each iteration	Overall Complexity
GD	$O\left(\exp\left(-\frac{t}{Q}\right)\right)$	$O(n \cdot d)$	$O\left(nd \cdot Q \cdot \log\left(\frac{1}{\epsilon}\right)\right)$	$O\left(\frac{\beta}{t}\right)$	$O(n \cdot d)$	$O\left(nd \cdot \beta \cdot \left(\frac{1}{\epsilon}\right)\right)$
SGD	$O\left(\frac{1}{t}\right)$	$O(d)$	$O\left(\frac{d}{\epsilon}\right)$	$O\left(\frac{1}{\sqrt{t}}\right)$	$O(d)$	$O\left(\frac{d}{\epsilon^2}\right)$

When data size n is very large, SGD is faster than GD.

Outline

- Optimization theory
- Generalization theory
- Approximation theory

Capacity of the Hypothesis Space

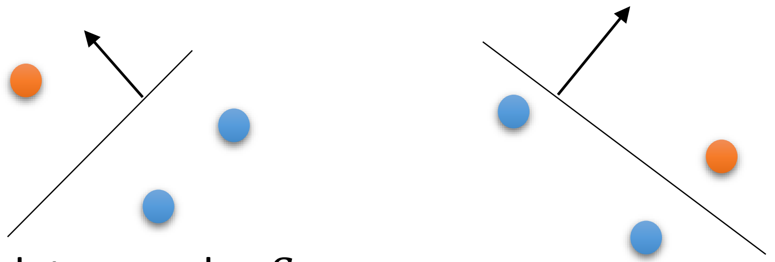
$$
\begin{aligned}
&Capacity\ (\mathcal{F}; n) \\
&= Complexity\ (\{f(x_1), \dots, f(x_n) : f \in \mathcal{F}, x_1, \dots x_n \in \mathcal{X}\}) \\
&= \ Complexity\left(\boxed{\mathcal{F}_{|S_n \in \mathcal{X}^n}}\right)
\end{aligned}
$$

$Projections\ of\ \mathcal{F}\ on\ finite\ data\ samples$

Example: $\mathcal{F} = \{linear\ classifiers\ on\ R^2\}$

Fix 2 data samples S_2

$\mathcal{F}_{|S_2} = \{(-1,1), (1,-1), (-1,-1), (1,1)\}$



Fix 3 data samples S_3

$\mathcal{F}_{|S_3} = \{(1,-1,-1), (-1,-1,1) \dots\}$

VC Dimension (Vapnik 1971)

Growth function:
If we measure $Complexity(\mathcal{F}_{|S_n \in \mathcal{X}^n})$ *by* $\max_{S_n \in \mathcal{X}^n} |\mathcal{F}_{|S_n \in \mathcal{X}^n}|$,
we call the corresponding capacity *growth function, denoted by* $G(\mathcal{F}, n)$

VC-dimension:
if $G(\mathcal{F}, n) = 2^n$, then the hypothesis space $\mathcal{F}$ can shatter n instances. If we measure $Complexity(\mathcal{F}_{|S_n \in \mathcal{X}^n})$ by the largest number of instances that $\mathcal{F}$ can shatter, i.e.,

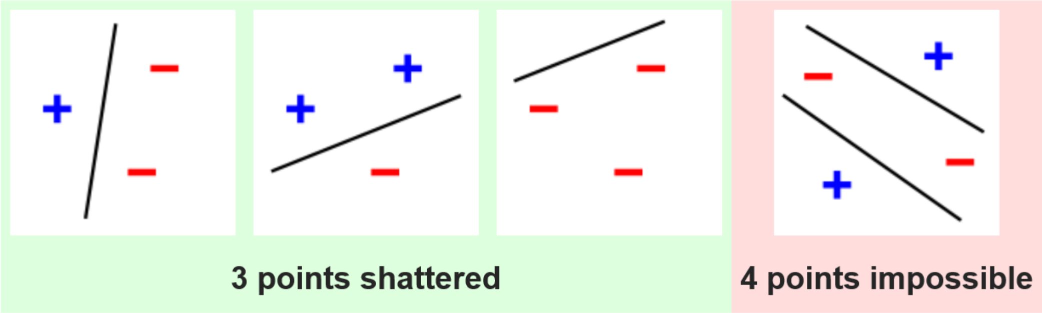
$$\mathrm{VC}(\mathcal{F}) = max\,\{n: G(\mathcal{F}, n) = 2^n\}$$

VC dimension v.s. Growth function:
(Sauer's Lemma) *Growth function can be upper bounded using VC dimension h,*

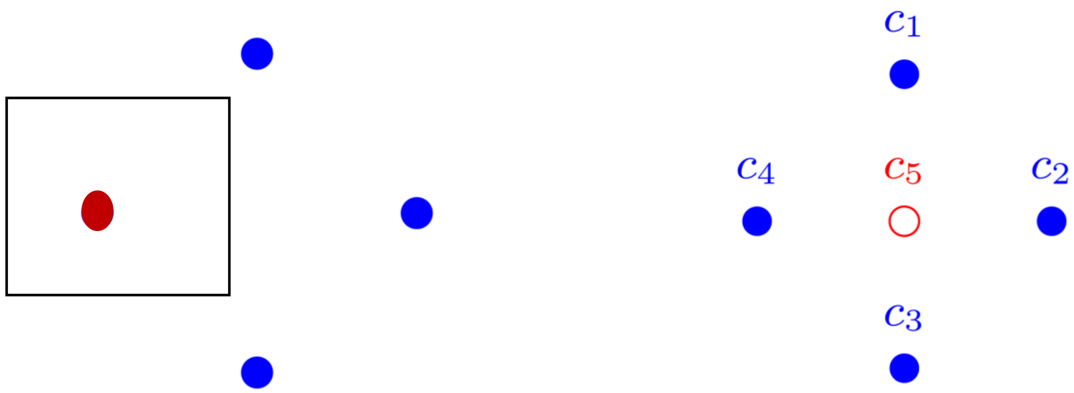
$$G(\mathcal{F}, n) \leq \sum_{i=0}^{h} \binom{n}{i} \leq \left(\frac{en}{d}\right)^d$$

Example $\mathrm{VC}(\mathcal{F}) = ?$

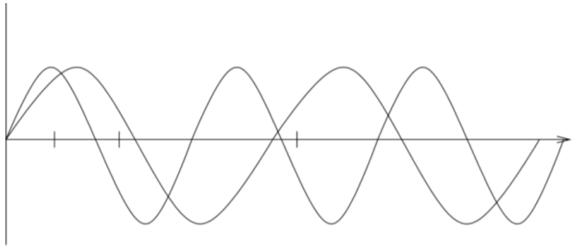
$$\mathcal{F} = \{linear\ classifiers\ on\ R^2\}$$



$$\mathcal{F} = \{axis\ aligned\ rectangels\ on\ R^2\}$$



$$\mathcal{F} = \{\mathrm{sgn}(\sin(tx)) : t \in R\}$$



VC Bound

Theorem 4: Assume the VC dimension of the hypothesis space $\mathcal{F}$ is h, then for arbitrary $n > h$ and $\delta > 0$, with probability at least $1 - \delta$, we have

$$\sup_{f \in \mathcal{F}} \left| \widehat{L_n}(f) - L(f) \right| \le \sqrt{\frac{8h\log\left(\frac{2en}{h}\right) + 8\log\frac{2}{\delta}}{n}}$$

$$L(f_n^*) - \inf_{f \in \mathcal{F}} L(f) \le 2 \sup_{f \in \mathcal{F}} \left| \widehat{L_n}(f) - L(f) \right| \le O\left(\sqrt{\frac{h}{n}}\right)$$

Covering Number

Covering number:

$N(X, \epsilon, d)\ with\ X \in R^m$ is the size of the ϵ-net of the set X with distance d over R^m.

Example (Covering number of Euclidean norm ball):

$$N(B(1), \epsilon, \|\cdot\|_2) \leq \left(1 + \frac{2}{\epsilon}\right)^m \leq \left(\frac{3}{\epsilon}\right)^m.$$

If we measure $Complexity\left(\mathcal{F}_{|S_n \in \mathcal{X}^n}\right)$ by $\max_{S_n \in \mathcal{X}^n} N(\mathcal{F}_{|S_n \in \mathcal{X}^n}, \epsilon, d)$, we call the corresponding capacity covering number, denoted as $\mathbf{N}(\boldsymbol{\mathcal{F}}; \mathbf{n}, \boldsymbol{\epsilon}, \mathbf{d})$**.** (Bartlett 1998)

Covering Number Bound

Theorem 5: for arbitrary $\epsilon > 0$, we have

$$\mathbb{P}\left\{\sup_{f\in\mathcal{F}}\left|\widehat{L_n}(f) - L(f)\right| \le \epsilon\right\} \ge 1 - 8\mathbb{E}[N(\mathcal{F}; n, \epsilon, d_H)]e^{-\frac{n\epsilon^2}{128}}.$$

Covering numbers can be defined for class of real-valued functions,
giving an estimation error bound for learning tasks other than classification.

Rademacher Average

Empirical Rademacher Average (Bartlett 2003):
If we measure $Complexity(\mathcal{F}_{|S_n \in \mathcal{X}^n})$ by the expected degree $\mathcal{F}_{|S_n \in \mathcal{X}^n}$ can fit the random noise sequence of length n, then we get the empirical Rademacher Average, i.e.,

$$RA(\mathcal{F}; S_n) = \frac{1}{n} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \sigma_i f(x_i) \right]$$

where $\sigma_1, \ldots, \sigma_n$ are independent uniform $\{\pm 1\}$-valued random variables.

Example: Suppose $S_n = \{x_1, \ldots, x_n\}$ and $x_i \in R^d$ for all $i \in [n]$

1. If $\|x_i\|_2 \le X_2$, $\mathcal{F} = \{\langle w, x \rangle; \|w\|_2 \le W_2\}$, then $RA(\mathcal{F}; S_n) \le \frac{X_2 W_2}{\sqrt{n}}$.
2. If $\|x\|_\infty \le X_\infty$, $\mathcal{F} = \{\langle w, x \rangle; \|w\|_1 \le W_1\}$, then $RA(\mathcal{F}; S_n) \le X_\infty W_1 \sqrt{\frac{2 \log d}{n}}$.

Rademacher Average

$$RA(\mathcal{F}; n) = \mathbb{E}_{S_n}[RA(\mathcal{F}; S_n)]$$

RA bound

Theorem 5: for arbitrary $\delta > 0$, we have, with probability at least $1 - \delta$,

$$\sup_{f \in \mathcal{F}} \left| \widehat{L_n}(f) - L(f) \right| \leq 2RA(\mathcal{F}; n) + \sqrt{\frac{\log \frac{1}{\delta}}{2n}}$$

RA v.s. Covering number v.s. VC:

$$RA(\mathcal{F}; n) \leq \frac{C}{\sqrt{n}} \int_0^\infty \sqrt{\log N(\mathcal{F}; n, \epsilon, d_H)} \leq C\sqrt{\frac{h}{n}}$$

Margin bound

- The test error can be upper bounded by the empirical margin loss and Rademacher Average divided by margin.
- Consider the prediction problem with $\mathcal{Y} \in \{-1, 1\}$ and define the margin as $f(x)y$, we have

$$L(f) \leq \hat{L}_\gamma(f) + \frac{RA(\mathcal{F}; n)}{\gamma} + O\left(\frac{1}{\sqrt{n}}\right),$$

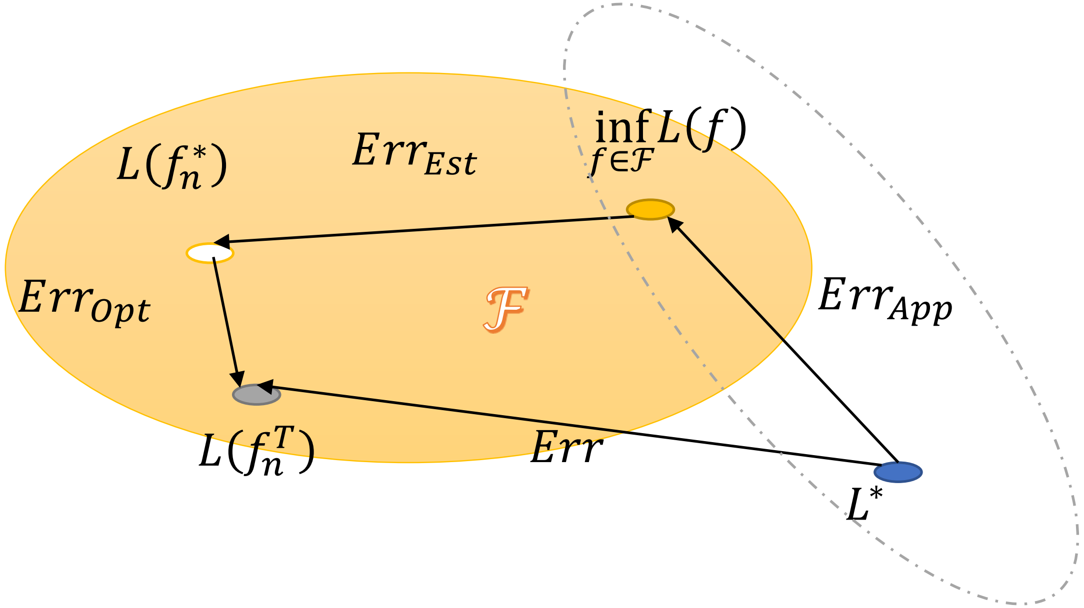
where $\hat{L}_\gamma(f) = \frac{1}{\mathrm{n}} \sum_i \ell_\gamma(y_i f(x_i)$ is the average margin loss and γ-margin loss

$$\ell_\gamma(t) = \begin{cases} 0, & \text{if } t \geq \gamma \\ 1, & \text{if } t \leq 0 \\ 1 - t/\gamma, & \text{otherwise} \end{cases}$$

Outline

- Optimization theory
- Generalization theory
- Approximation theory

Approximation Error



Continuous function on compact set.

Assume $L^* = L(\boxed{f^*})$,

if $\exists f_m \in \boxed{\mathcal{F}}$ and $\boxed{f_m \to f^*}$,

then $\inf_{f \in \mathcal{F}} L(f) - L^* = 0$

L_∞- convergence

2-layer neural networks with finite hidden units

Universal Approximation of Neural Networks

- (Hornik 1989) Feedforward networks with only a single hidden layer can approximate any continuous function **uniformly** on any compact set and any measurable function arbitrarily well.
- For example, $\forall\ f \in C\left([0,1]^d\right), \forall \epsilon > 0, \exists$ 2-layer neural network $NN, s.t.$

$$\forall x \in [0,1]^d, |NN(x) - f(x)| \leq \epsilon.$$

Overall Picture of SLT

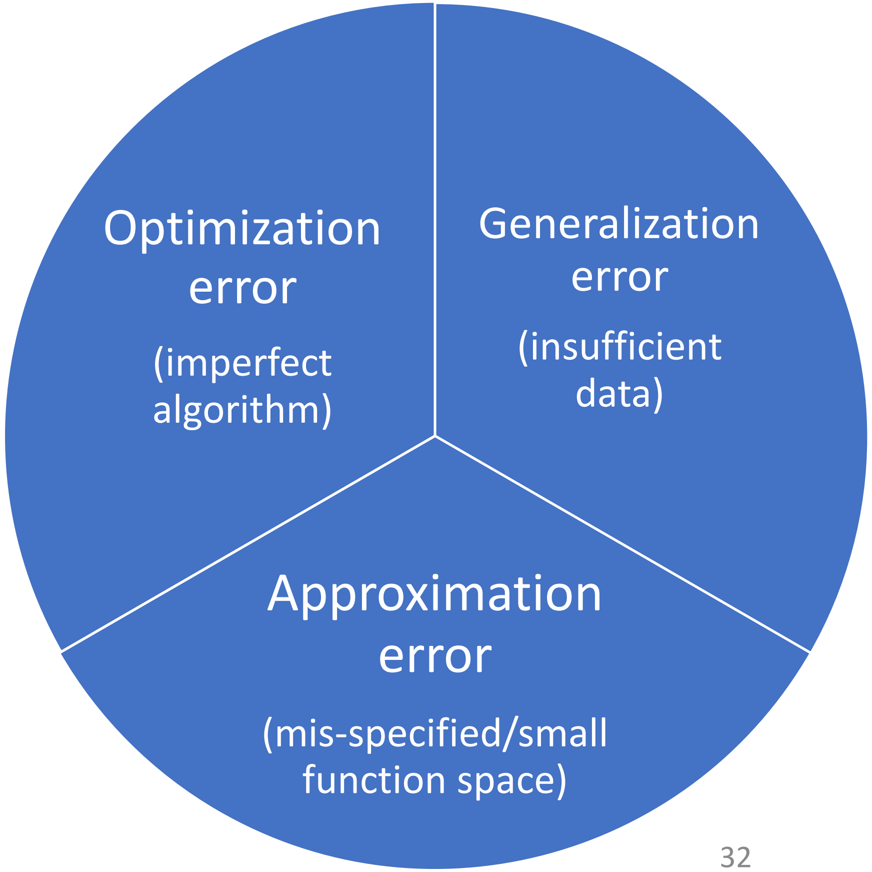
- Training: Find a function f from a function class $\mathcal{F}$ based on training dataset $\mathcal{D}$.

$$\arg\min_{f \in \boldsymbol{\mathcal{F}}} \mathbb{E}_{(x_i, y_i) \in \mathcal{D}} \boldsymbol{L}(f(x_i), y_i)$$

- Evaluation: How does f perform on test data: good or not?

$$\mathbb{E}_{(x_i, y_i) \in \boldsymbol{\mathcal{P}}} \boldsymbol{L}(f(x_i), y_i)$$

- Where is the gap?
  - $argmin$: optimization error → convergence of the algorithm
  - $\mathcal{D} \rightarrow \mathcal{P}$: generalization error → hypothesis space capacity
  - Hypothesis space $\mathcal{F}$: approximation error → hypothesis space capacity



Reference

- 周志华，机器学习，清华出版社
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press
- Vapnik, The Nature of Statistical Learning Theory, Springer, 1999
- Yoav Freund and Robert E. Schapire, A Short Introduction to Boosting, 1999.
- Shalev-Shwartz, Shai, and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- Bousquet, Olivier, Stéphane Boucheron, and Gábor Lugosi. "Introduction to statistical learning theory." Springer, Berlin, Heidelberg, 2003.
- Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators[J]. Neural networks, 1989, 2(5): 359-366.

Thanks!

tyliu@microsoft.com

http://research.microsoft.com/users/tyliu/