Machine Learning
(機器學習)

Lecture 07: Combatting Overfitting

Hsuan-Tien Lin (林軒田)

htlin@csie.ntu.edu.tw

Department of Computer Science
& Information Engineering

National Taiwan University
(國立台灣大學資訊工程系)

Roadmap

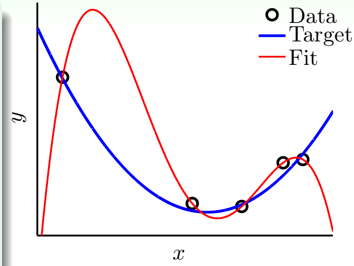
1. When Can Machines Learn?
2. Why Can Machines Learn?
3. How Can Machines Learn?
4. How Can Machines Learn **Better**?

Lecture 07: Combatting Overfitting

- What is Overfitting?
- The Role of Noise and Data Size
- Deterministic Noise
- Dealing with Overfitting
- Regularized Hypothesis Set
- Weight Decay Regularization
- Regularization and VC Theory
- General Regularizers

Bad Generalization

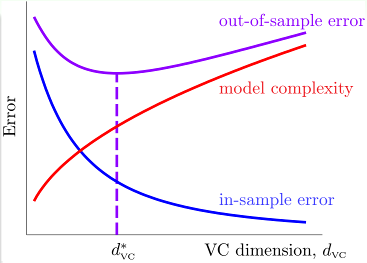
- regression for $x \in \mathbb{R}$ with $N = 5$ examples
- target $f(x)$ = 2nd order polynomial
- label $y_n = f(x_n) +$ very small noise
- linear regression in $\mathcal{Z}$-space + $\mathbf{\Phi}$ = 4th order polynomial
- unique solution passing all examples $\Longrightarrow E_{\text{in}}(g) = 0$
- $E_{\text{out}}(g)$ **huge**



bad generalization: low E_{in}, high E_{out}

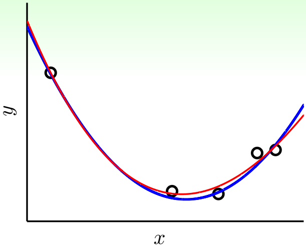
Bad Generalization and Overfitting

- take $d_{\mathrm{VC}} = 1126$ for learning: bad generalization —(E_{out} - E_{in}) large
- switch from $d_{\mathrm{VC}} = d^*_{\mathrm{VC}}$ to $d_{\mathrm{VC}} = 1126$: **overfitting** —$E_{\mathrm{in}} \downarrow$, $E_{\mathrm{out}} \uparrow$
- switch from $d_{\mathrm{VC}} = d^*_{\mathrm{VC}}$ to $d_{\mathrm{VC}} = 1$: **underfitting** —$E_{\mathrm{in}} \uparrow$, $E_{\mathrm{out}} \uparrow$

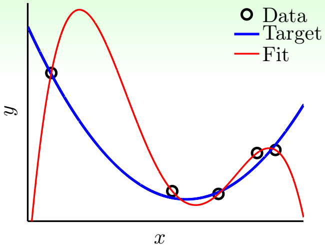


bad generalization: low E_{in}, high E_{out};
overfitting: low**er** E_{in}, high**er** E_{out}

Cause of Overfitting: A Driving Analogy



'good fit' $\Longrightarrow$ **overfit**



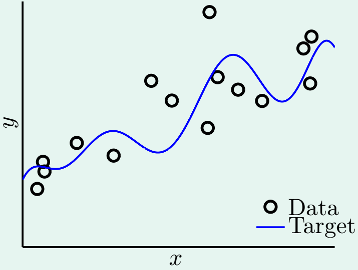
learning	driving
overfit	commit a car accident
use excessive d_{VC}	'drive too fast'
noise	bumpy road
limited data size N	limited observations about road condition

next: how does **noise** & **data size** affect overfitting?

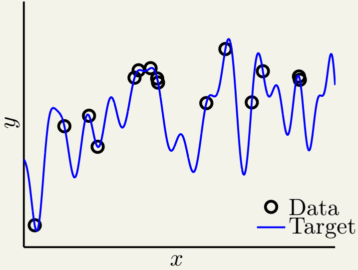
Questions?

Case Study (1/2)

10-th order target function + noise



50-th order target function noiselessly



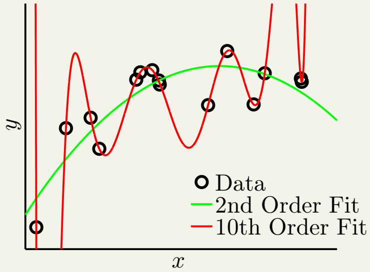
overfitting from best $g_2 \in \mathcal{H}_2$ to best $g_{10} \in \mathcal{H}_{10}$?

Case Study (2/2)

10-th order target function + noise

	$g_2 \in \mathcal{H}_2$	$g_{10} \in \mathcal{H}_{10}$
E_{in}	0.050	0.034
E_{out}	0.127	**9.00**

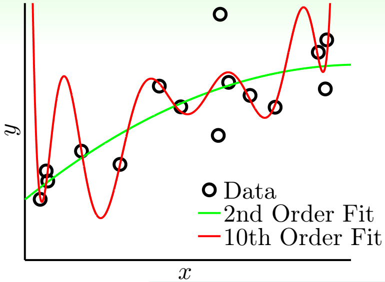
50-th order target function noiselessly

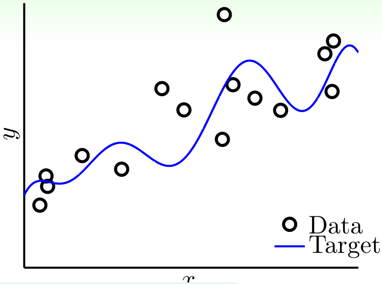


	$g_2 \in \mathcal{H}_2$	$g_{10} \in \mathcal{H}_{10}$
E_{in}	0.029	0.00001
E_{out}	0.120	**7680**

overfitting from g_2 to g_{10}? **both yes!**

Irony of Two Learners



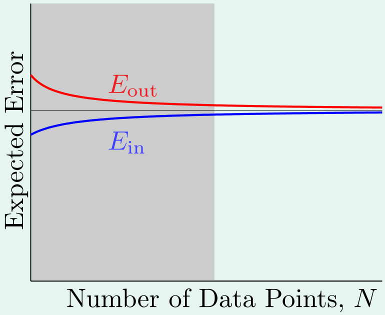


- learner *Overfit*: pick $g_{10} \in \mathcal{H}_{10}$
- learner *Restrict*: pick $g_2 \in \mathcal{H}_2$
- when both **know that target = 10th**
  —*R* 'gives up' ability to fit

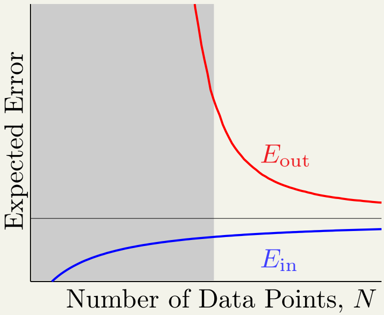
but *R* **wins in** E_{out} a lot!
philosophy: concession for **advantage**? **:-)**

Learning Curves Revisited

$\mathcal{H}_2$



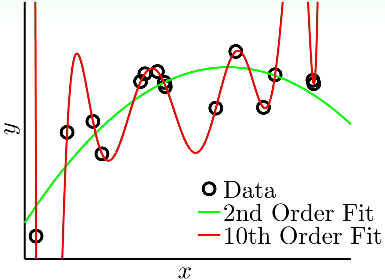
$\mathcal{H}_{10}$

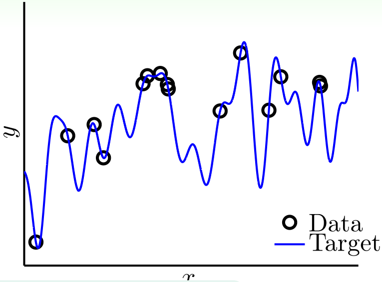


- $\mathcal{H}_{10}$: lower $\overline{E_{\text{out}}}$ when $N \to \infty$, but much larger generalization error for small N
- gray area : O overfits! ($\overline{E_{\text{in}}} \downarrow$, $\overline{E_{\text{out}}} \uparrow$)

R always **wins in $\overline{E_{\text{out}}}$** if N small!

The ‘No Noise’ Case





- learner *Overfit*: pick $g_{10} \in \mathcal{H}_{10}$
- learner *Restrict*: pick $g_2 \in \mathcal{H}_2$
- when both **know that there is no noise** —*R* still wins

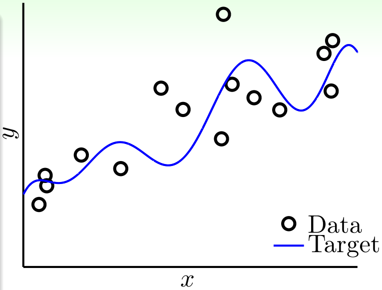
is there really **no noise**?
‘target complexity’ acts like noise

Questions?

A Detailed Experiment

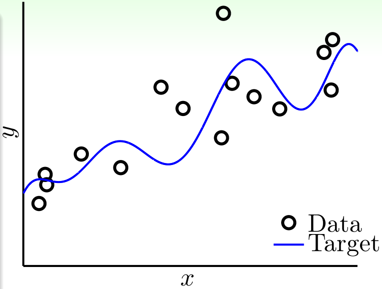
$$y = f(x) + \epsilon \sim \text{Gaussian}\left(\underbrace{\sum_{q=0}^{Q_f} \alpha_q x^q}_{f(x)}, \sigma^2\right)$$

- Gaussian iid noise ϵ with level σ^2
- some 'uniform' distribution on $f(x)$ with complexity level Q_f
- data size N



goal: **'overfit level'** for different (N, σ^2) and (N, Q_f)?

The Overfit Measure

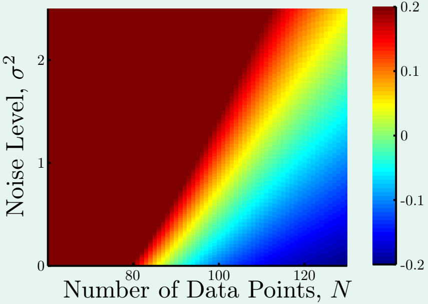


- $g_2 \in \mathcal{H}_2$
- $g_{10} \in \mathcal{H}_{10}$
- $E_{\text{in}}(g_{10}) \le E_{\text{in}}(g_2)$ for sure

overfit measure $E_{\text{out}}(g_{10}) - E_{\text{out}}(g_2)$

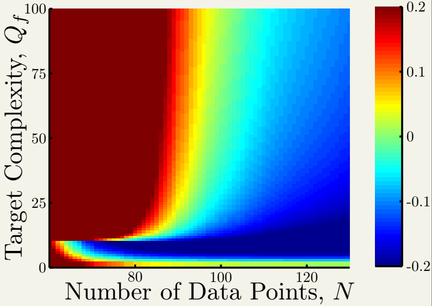
The Results

impact of σ^2 versus N



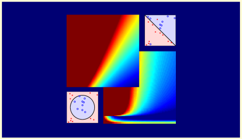
fixed $Q_f = 20$

impact of Q_f versus N



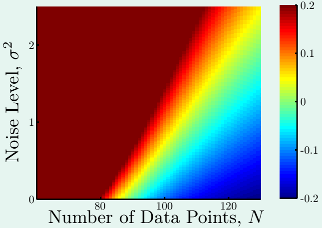
fixed $\sigma^2 = 0.1$

ring a bell? :-)

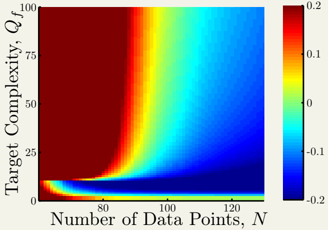


Impact of Noise and Data Size

impact of σ^2 versus N: **stochastic noise**



impact of Q_f versus N: **deterministic noise**



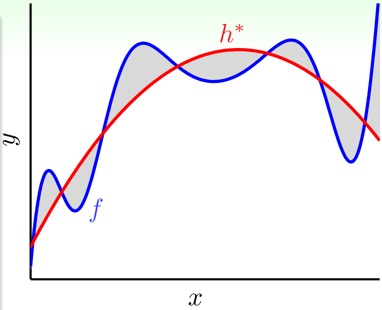
four reasons of serious overfitting:

data size $N \downarrow$	overfit $\uparrow$
stochastic noise $\uparrow$	overfit $\uparrow$
deterministic noise $\uparrow$	overfit $\uparrow$
excessive power $\uparrow$	overfit $\uparrow$

overfitting 'easily' happens

Deterministic Noise

- if $f \notin \mathcal{H}$: something of f cannot be captured by $\mathcal{H}$
- deterministic noise : difference between best $h^* \in \mathcal{H}$ and f
- acts like 'stochastic noise'—not new to CS: pseudo-random generator
- difference to stochastic noise:
  - depends on $\mathcal{H}$
  - fixed for a given $\mathbf{x}$



philosophy: when teaching a kid,
perhaps better not to use examples
from a complicated target function? :-)

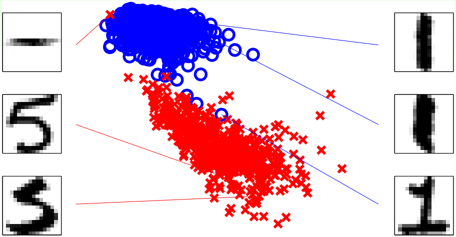
Questions?

Driving Analogy Revisited

learning	driving
overfit	commit a car accident
use excessive d_{VC}	'drive too fast'
noise	bumpy road
limited data size N	limited observations about road condition
start from simple model	drive slowly
data cleaning/pruning	use more accurate road information
data hinting	exploit more road information
regularization	put the brakes
validation	monitor the dashboard

all very **practical** techniques
to combat overfitting

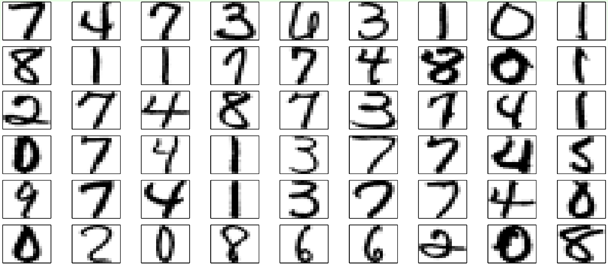
Data Cleaning/Pruning



- if 'detect' the outlier 5 at the top by
  - too close to other ∘, or too far from other ×
  - wrong by current classifier
  - . . .
- possible action 1: correct the label (**data cleaning**)
- possible action 2: remove the example (**data pruning**)

possibly helps, but **effect varies**

Data Hinting

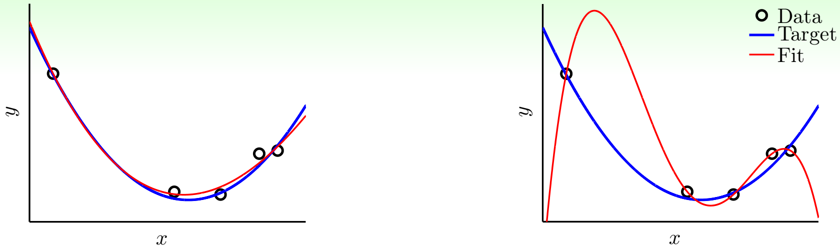


- slightly shifted/rotated digits carry the same meaning
- possible action: add **virtual examples** by shifting/rotating the given digits (**data hinting**, **data augmentation**)

possibly helps, but **watch out**

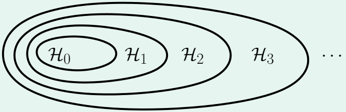
—**virtual example not** $\overset{iid}{\sim} P(\mathbf{x}, y)$!

Regularization: The Magic of 'Brake'



'regularized fit' $\Longleftarrow$ overfit

- idea: 'step back' from $\mathcal{H}_{10}$ to $\mathcal{H}_2$

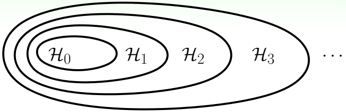


- name history: function approximation for **ill-posed problems**

how to step back?

Questions?

Stepping Back as Constraint



Q-th order polynomial transform for $x \in \mathbb{R}$:

$$\boldsymbol{\Phi}_Q(x) = \left(1, x, x^2, \ldots, x^Q\right)$$

+ linear regression, denote $\tilde{\mathbf{w}}$ by $\mathbf{w}$

hypothesis $\mathbf{w}$ in $\mathcal{H}_{10}$: $w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_{10} x^{10}$

hypothesis $\mathbf{w}$ in $\mathcal{H}_2$: $w_0 + w_1 x + w_2 x^2$

that is, $\mathcal{H}_2 = \mathcal{H}_{10}$ AND 'constraint that $w_3 = w_4 = \ldots = w_{10} = 0$'

step back = **constraint**

Regression with Constraint

$$\mathcal{H}_{10} \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \right\}$$

regression with $\mathcal{H}_{10}$:

$$\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w})$$

$$\mathcal{H}_2 \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \text{ while } w_3 = w_4 = \ldots = w_{10} = 0 \right\}$$

regression with $\mathcal{H}_2$:

$$\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w}) \quad \text{s.t. } w_3 = w_4 = \ldots = w_{10} = 0$$

step back = constrained optimization of E_{in}

why don't you just use $\mathbf{w} \in \mathbb{R}^{2+1}$? **:-)**

Regression with Looser Constraint

$$\mathcal{H}_2 \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \text{ while } w_3 = \ldots = w_{10} = 0 \right\}$$

regression with $\mathcal{H}_2$:

$$\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w}) \quad \text{s.t.} \quad w_3 = \ldots = w_{10} = 0$$

$$\mathcal{H}'_2 \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \text{ while } \geq 8 \text{ of } w_q = 0 \right\}$$

regression with $\mathcal{H}'_2$:

$$\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w}) \quad \text{s.t.} \quad \sum_{q=0}^{10} [\![w_q \neq 0]\!] \leq 3$$

- more flexible than $\mathcal{H}_2$: $\mathcal{H}_2 \subset \mathcal{H}'_2$
- less risky than $\mathcal{H}_{10}$: $\mathcal{H}'_2 \subset \mathcal{H}_{10}$

bad news for sparse hypothesis set $\mathcal{H}'_2$:
NP-hard to solve :-(

Regression with Softer Constraint

$$\mathcal{H}'_2 \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \text{ while } \geq 8 \text{ of } w_q = 0 \right\}$$

regression with $\mathcal{H}'_2$:

$$\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w}) \text{ s.t. } \sum_{q=0}^{10} [\![w_q \neq 0]\!] \leq 3$$

$$\mathcal{H}(C) \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \text{ while } \|\mathbf{w}\|^2 \leq C \right\}$$

regression with $\mathcal{H}(C)$:

$$\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w}) \text{ s.t. } \sum_{q=0}^{10} w_q^2 \leq C$$

- $\mathcal{H}(C)$: overlaps but not exactly the same as $\mathcal{H}'_2$
- soft and smooth structure over $C \geq 0$:
  $\mathcal{H}(0) \subset \mathcal{H}(1.126) \subset \ldots \subset \mathcal{H}(1126) \subset \ldots \subset \mathcal{H}(\infty) = \mathcal{H}_{10}$

regularized hypothesis $\mathbf{w}_{\text{REG}}$:
optimal solution from
regularized hypothesis set $\mathcal{H}(C)$

Questions?

Matrix Form of Regularized Regression Problem

$$\min_{\mathbf{w}\in\mathbb{R}^{Q+1}} \quad E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\underbrace{\sum_{n=1}^{N}(\mathbf{w}^T\mathbf{z}_n - y_n)^2}_{(\mathrm{Z}\mathbf{w}-\mathbf{y})^T(\mathrm{Z}\mathbf{w}-\mathbf{y})}$$

$$\text{s.t.} \quad \underbrace{\sum_{q=0}^{Q} w_q^2}_{\mathbf{w}^T\mathbf{w}} \leq C$$

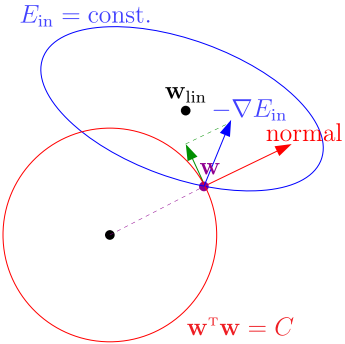
- $\sum_n \ldots = (\mathrm{Z}\mathbf{w}-\mathbf{y})^T(\mathrm{Z}\mathbf{w}-\mathbf{y})$, **remember? :-)**
- $\mathbf{w}^T\mathbf{w} \leq C$: feasible $\mathbf{w}$ within a radius-$\sqrt{C}$ hypersphere

how to solve
constrained optimization problem?

The Lagrange Multiplier

$$\min_{\mathbf{w} \in \mathbb{R}^{Q+1}} \quad E_{\text{in}}(\mathbf{w}) = \frac{1}{N}(\mathrm{Z}\mathbf{w} - \mathbf{y})^T(\mathrm{Z}\mathbf{w} - \mathbf{y}) \text{ s.t. } \mathbf{w}^T\mathbf{w} \le C$$

- decreasing direction: $-\nabla E_{\text{in}}(\mathbf{w})$, **remember? :-)**
- normal vector of $\mathbf{w}^T\mathbf{w} = C$: $\mathbf{w}$
- if $-\nabla E_{\text{in}}(\mathbf{w})$ and $\mathbf{w}$ not parallel: can **decrease $E_{\text{in}}(\mathbf{w})$ without violating the constraint**
- at optimal solution $\mathbf{w}_{\text{REG}}$, $-\nabla E_{\text{in}}(\mathbf{w}_{\text{REG}}) \propto \boxed{\mathbf{w}_{\text{REG}}}$



want: find Lagrange multiplier $\lambda > 0$ and $\mathbf{w}_{\text{REG}}$ such that $\nabla E_{\text{in}}(\mathbf{w}_{\text{REG}}) + \frac{2\lambda}{N}\boxed{\mathbf{w}_{\text{REG}}} = \mathbf{0}$

Augmented Error

- if oracle tells you $\lambda > 0$, then

solving
$$\nabla E_{\text{in}}(\mathbf{w}_{\text{REG}}) + \frac{2\lambda}{N}\boxed{\mathbf{w}_{\text{REG}}} = \mathbf{0}$$

$$\frac{2}{N}\left(\mathrm{Z}^T\mathrm{Z}\mathbf{w}_{\text{REG}} - \mathrm{Z}^T\mathbf{y}\right) + \frac{2\lambda}{N}\boxed{\mathbf{w}_{\text{REG}}} = \mathbf{0}$$

- optimal solution:

$$\mathbf{w}_{\text{REG}} \leftarrow (\mathrm{Z}^T\mathrm{Z} + \lambda\mathrm{I})^{-1}\mathrm{Z}^T\mathbf{y}$$

—called ridge regression in Statistics

minimizing unconstrained E_{aug} effectively
minimizes some C-constrained E_{in}

Augmented Error

- if oracle tells you $\lambda > 0$, then

$$\text{solving} \qquad \nabla E_{\text{in}}(\mathbf{w}_{\text{REG}}) + \frac{2\lambda}{N} \boxed{\mathbf{w}_{\text{REG}}} = \mathbf{0}$$

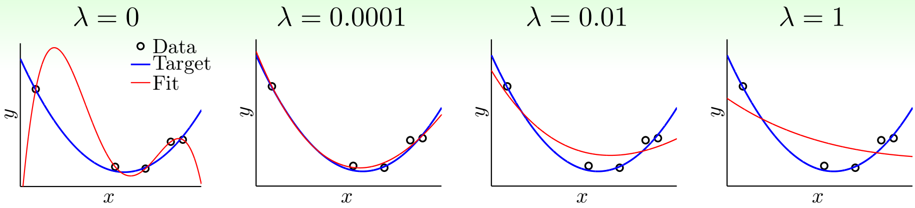
$$\text{equivalent to minimizing} \qquad \underbrace{E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N} \overbrace{\mathbf{w}^T \mathbf{w}}^{\textbf{regularizer}}}_{\textbf{augmented error } E_{\text{aug}}(\mathbf{w})}$$

- regularization with augmented error instead of constrained E_{in}

$$\mathbf{w}_{\text{REG}} \leftarrow \underset{\mathbf{w}}{\operatorname{argmin}}\, E_{\text{aug}}(\mathbf{w}) \text{ for given } \lambda > 0 \text{ or } \lambda = 0$$

minimizing unconstrained E_{aug} effectively
minimizes some C-constrained E_{in}

The Results



overfitting $\Longrightarrow$ $\Longrightarrow$ $\Longrightarrow$ underfitting

philosophy: *a little **regularization** goes a long way!*

call '$+\frac{\lambda}{N}\mathbf{w}^T\mathbf{w}$' **weight-decay** regularization:

larger λ
$\Longleftrightarrow$ prefer shorter $\mathbf{w}$
$\Longleftrightarrow$ effectively smaller C

—go with 'any' transform + linear model

Questions?

Regularization and VC Theory

Regularization by Constrained-Minimizing E_{in}

$$\min_{\mathbf{w}} E_{\text{in}}(\mathbf{w}) \text{ s.t. } \mathbf{w}^T\mathbf{w} \le C$$

$\longrightarrow$

VC Guarantee of Constrained-Minimizing E_{in}

$$E_{\text{out}}(\mathbf{w}) \le E_{\text{in}}(\mathbf{w}) + \Omega(\mathcal{H}(C))$$

$\Updownarrow$ C equivalent to some λ

Regularization by Minimizing E_{aug}

$$\min_{\mathbf{w}} E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\mathbf{w}^T\mathbf{w}$$

minimizing E_{aug}: indirectly getting VC guarantee **without confining to $\mathcal{H}(C)$**

Another View of Augmented Error

Augmented Error

$$E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\mathbf{w}^T\mathbf{w}$$

VC Bound

$$E_{\text{out}}(\mathbf{w}) \le E_{\text{in}}(\mathbf{w}) + \Omega(\mathcal{H})$$

- regularizer $\mathbf{w}^T\mathbf{w}$: complexity of a single hypothesis
- generalization price $\Omega(\mathcal{H})$: complexity of a hypothesis set
- if $\frac{\lambda}{N}\Omega(\mathbf{w})$ 'represents' $\Omega(\mathcal{H})$ well,
  E_{aug} is a better proxy of E_{out} than E_{in}

minimizing E_{aug}:

(heuristically) operating with the better proxy;
(technically) enjoying flexibility of whole $\mathcal{H}$

Effective VC Dimension

$$\min_{\mathbf{w}\in\mathbb{R}^{\tilde{d}+1}} E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\Omega(\mathbf{w})$$

- model complexity?
  $d_{\text{VC}}(\mathcal{H}) = \tilde{d} + 1$, because $\{\mathbf{w}\}$ '**all considered**' during minimization
- $\{\mathbf{w}\}$ '**actually needed**': $\mathcal{H}(C)$, with some C equivalent to λ
- $d_{\text{VC}}\big(\mathcal{H}(C)\big)$:
  effective VC dimension $d_{\text{EFF}}(\mathcal{H}, \underbrace{\mathcal{A}}_{\min E_{\text{aug}}})$

explanation of regularization:
$d_{\text{VC}}(\mathcal{H})$ large,
while $d_{\text{EFF}}(\mathcal{H}, \mathcal{A})$ small if $\mathcal{A}$ regularized

Questions?

General Regularizers $\Omega(\mathbf{w})$

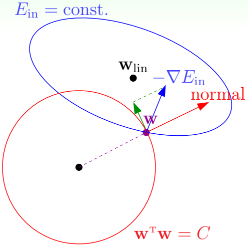
want: constraint in the **'direction' of target function**

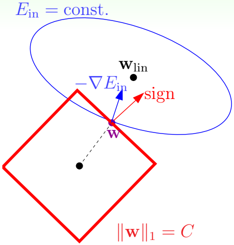
- target-dependent: some properties of target, if known
  - symmetry regularizer: $\sum [\![q \text{ is odd}]\!] w_q^2$
- plausible: direction towards smoother or simpler

  stochastic/deterministic noise both **non-smooth**
  - sparsity (L1) regularizer: $\sum |w_q|$ (next slide)
- friendly: easy to optimize
  - weight-decay (L2) regularizer: $\sum w_q^2$
- bad? **:-)**: no worries, guard by λ

augmented error = error $\widehat{\text{err}}$ + regularizer Ω
regularizer: target-dependent, plausible, or friendly

L2 and L1 Regularizer





L2 Regularizer

$$\Omega(\mathbf{w}) = \sum_{q=0}^{Q} w_q^2 = \|\mathbf{w}\|_2^2$$

- convex, differentiable everywhere
- easy to optimize

L1 Regularizer

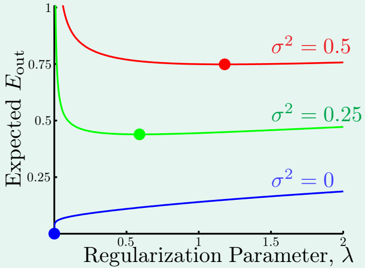
$$\Omega(\mathbf{w}) = \sum_{q=0}^{Q} |w_q| = \|\mathbf{w}\|_1$$

- convex, **not** differentiable everywhere
- **sparsity** in solution

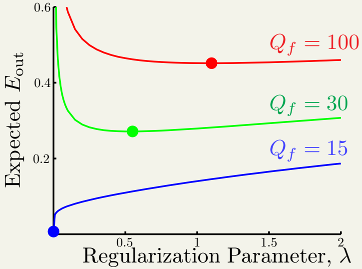
L1 useful if needing **sparse solution**

The Optimal λ

stochastic noise



deterministic noise



- more noise $\Longleftrightarrow$ more regularization needed
  —more bumpy road $\Longleftrightarrow$ putting brakes more
- noise **unknown**—important to **make proper choices**

how to choose?
stay tuned for the next lecture! :-)

Questions?

Summary

1. How Can Machines Learn?

Lecture 06: Beyond Basic Linear Models

2. How Can Machines Learn **Better**?

Lecture 07: Combatting Overfitting

- What is Overfitting?
  lower E_{in} but higher E_{out}
- The Role of Noise and Data Size
  overfitting 'easily' happens!
- Deterministic Noise
  what $\mathcal{H}$ cannot capture acts like noise
- Dealing with Overfitting
  data cleaning/pruning/hinting & regularization
- Regularized Hypothesis Set
  original $\mathcal{H}$ + constraint
- Weight Decay Regularization
  add $\frac{\lambda}{N}\mathbf{w}^T\mathbf{w}$ in E_{aug}
- Regularization and VC Theory
  regularization decreases d_{EFF}
- General Regularizers
  target-dependent, [plausible], or [friendly]

- **next: choosing from the so-many models/parameters**