Regularization and Validation

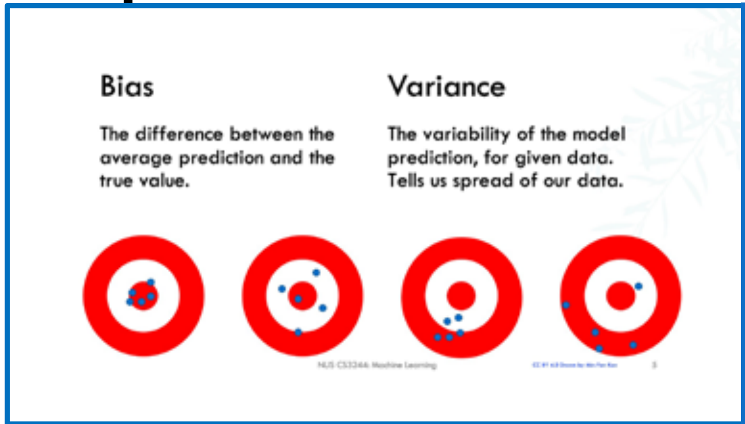
6

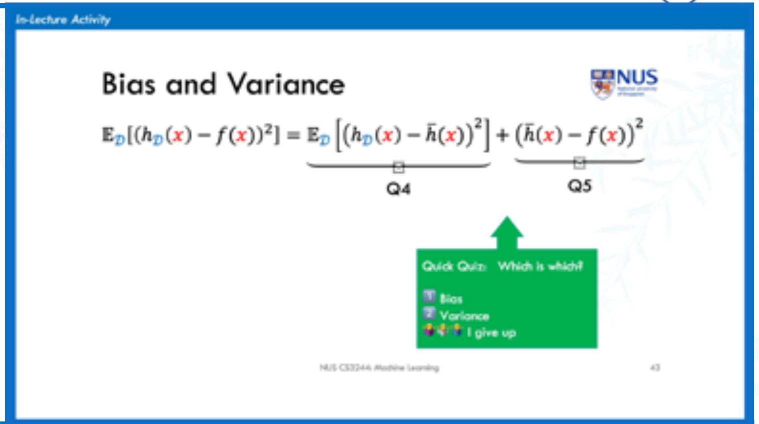
A

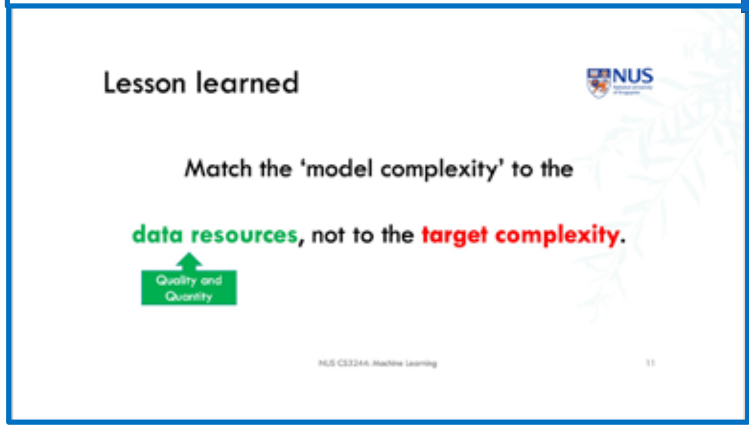
CS 3244
Machine Learning

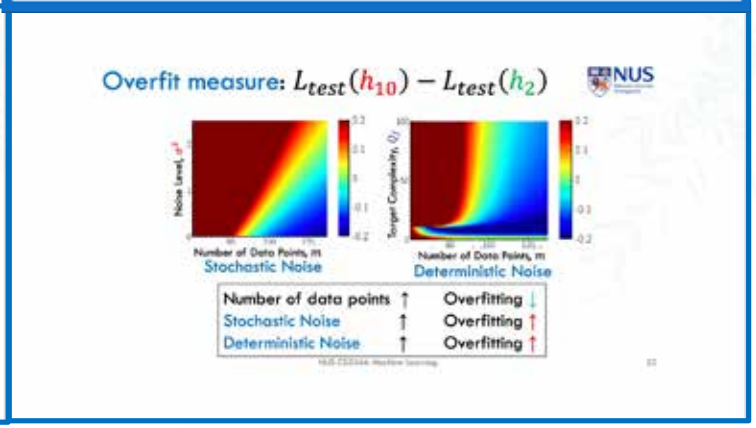
NUS National University of Singapore | Computing

Recap from Week 05









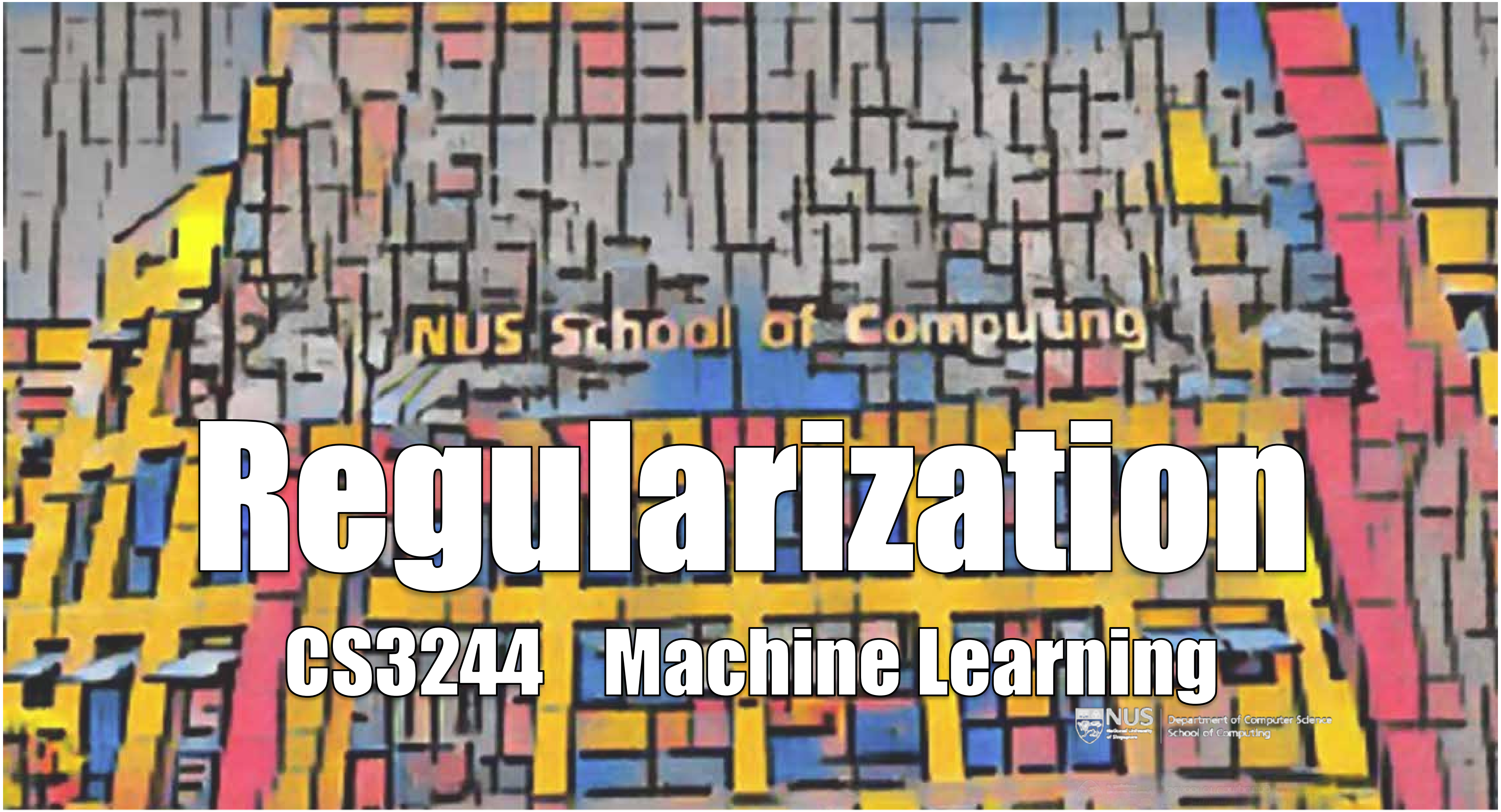
Forecast for Week 06

Learning Outcomes for this week:

- Understand **Regularization** as a means of restraining the model.
- Choose appropriate doses of regularization for a model.
- Understand and execute **Validation**, as a reality check by peeking (at the bottom line).
- Understand the different forms extending validation to encompass additional estimation.
- Understand how validation and regularization complement each other and their roles in affecting learning.

Regularization

CS3244 Machine Learning



Two Cures

In one form or another, $L_{test}(h) = L_{train}(h) +$ overfit penalty

1. **Regularization**: Restrain the model

$$L_{test}(h) = L_{train}(h) + \text{overfit penalty}$$

Regularization estimates this quantity

2. **Validation**: Reality check by peeking (at the bottom line)

$$L_{test}(h) = L_{train}(h) + \text{overfit penalty}$$

Restraining the model

Regularization

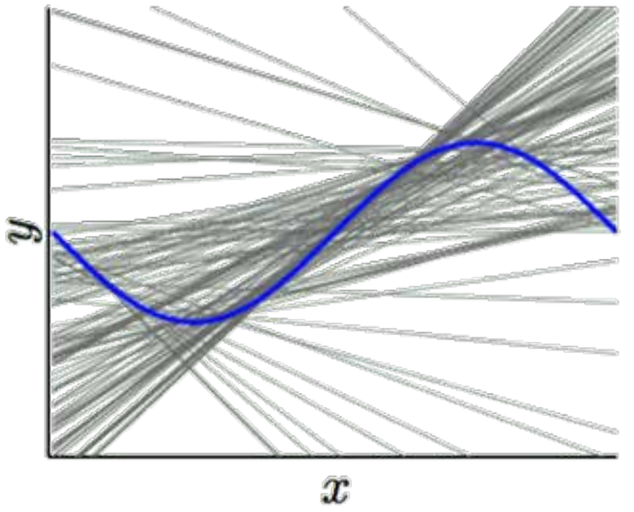
- What is it: A cure for our tendency to fit (get distracted by) the noise, hence improving L_{test}.

- How does it work?
  By constraining the model so that we cannot fit the noise.

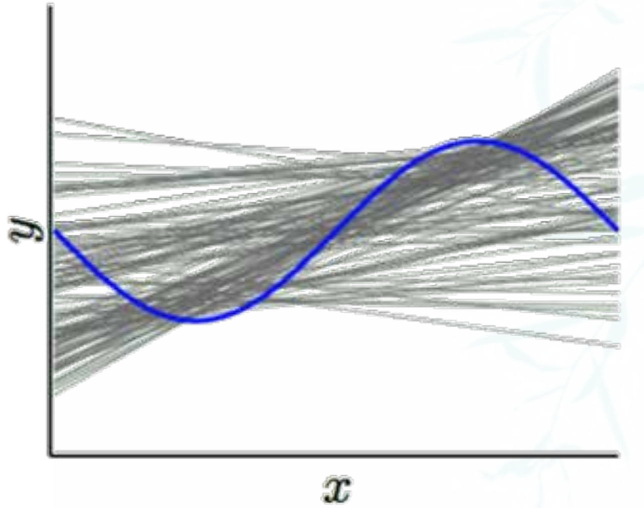
- Side effect: if we cannot fit the noise, maybe we cannot fit the signal f.

A familiar example

Without regularization



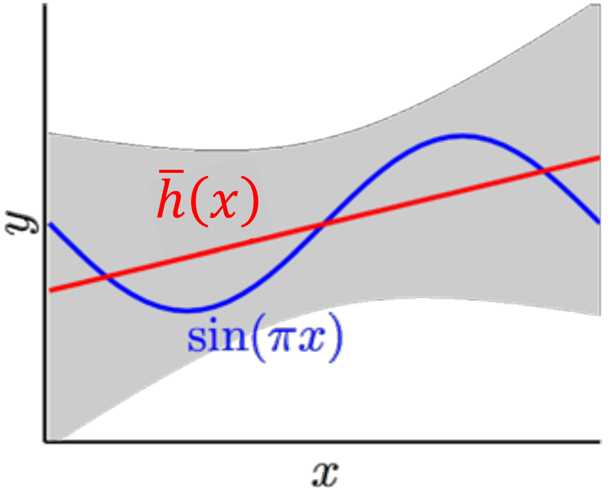
With regularization



Constrain weights to be a bit smaller

Bias goes up a little

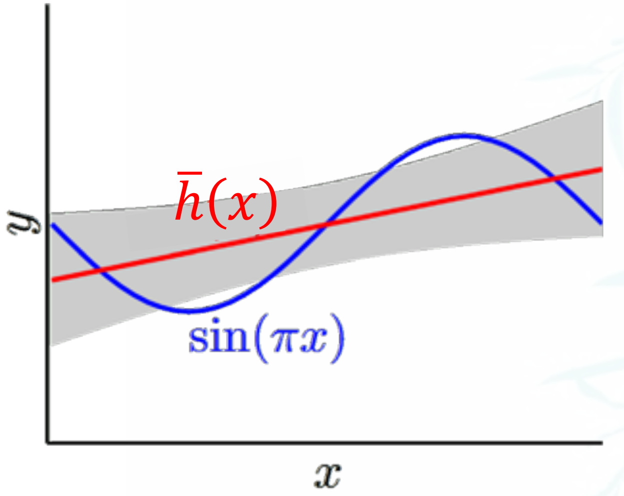
Without regularization



Bias = 0.21

Variance = 1.69

With regularization



Bias = 0.23

Variance = 0.33

Side Effect

Constraining the model: $\mathcal{H}_2$ versus $\mathcal{H}_{10}$

$$\mathcal{H}_{10} = \{h(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}\}$$

$$\mathcal{H}_2 = \{h(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}\}$$

such that $\theta_3 = \theta_4 = \cdots = \theta_{10} = 0$

A "hard" order constraint that sets some weights to zero.

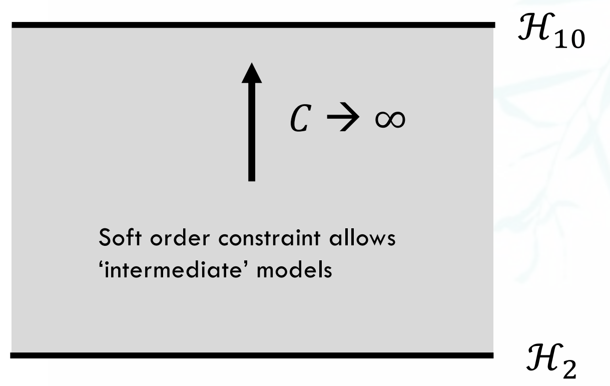
$$\mathcal{H}_2 \subset \mathcal{H}_{10}$$

Soft Order Constraint

Don't set weights explicitly to zero.

Re- use loss optimization by giving a budget and let the learning choose. Introduce a regularization function $\Omega(h)$.

$$\Omega(h) \equiv \sum_{q=0}^{Q} \theta_q^2 \leq C$$



Soft Order Constrained Model

$$\mathcal{H}_{10} = \{h(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}\}$$

$\mathcal{H}$ is larger

$$\mathcal{H}_C = \{h(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}\}$$

such that $\sum_{q=0}^{10} \theta_q^2 \leq C$

N.B. $\mathcal{H}_2$ and C=2 are unrelated!

$$\mathcal{H}_2 = \{h(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}\}$$

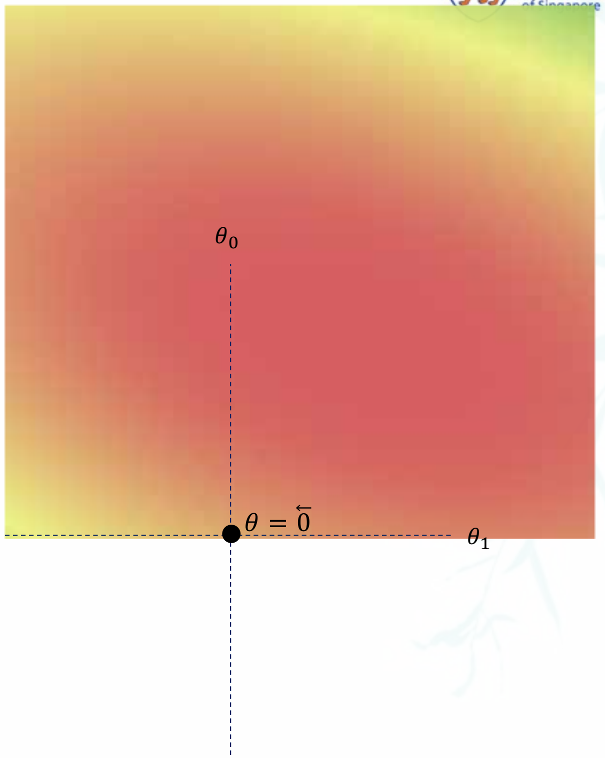
such that $\theta_3 = \theta_4 = \cdots = \theta_{10} = 0$

$\mathcal{H}$ is smaller

Solving for $\boldsymbol{\theta}_{reg}$

Minimize $L_{train} = \frac{1}{m}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^\top(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})$

Subject to a regularization function $\Omega(h) \equiv \boldsymbol{\theta}^\top\boldsymbol{\theta} \leq C$



Solving for $\boldsymbol{\theta}_{reg}$

Minimize $L_{train} = \frac{1}{m}(\mathbf{X\theta} - \mathbf{y})^\top(\mathbf{X\theta} - \mathbf{y})$

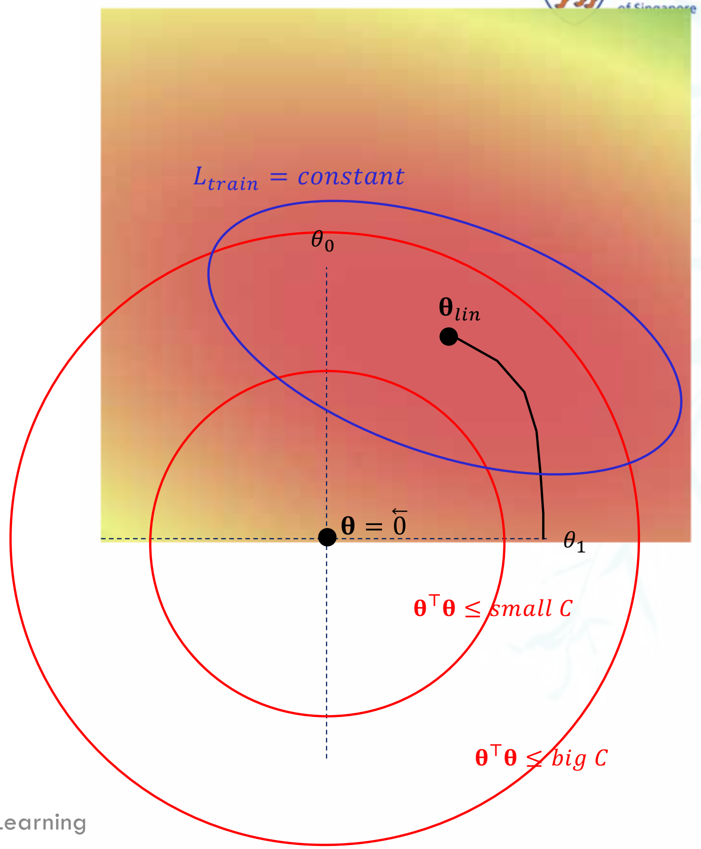
Subject to $\boldsymbol{\theta}^\top\boldsymbol{\theta} \leq C$

<u>Pictorially with 2 weights :</u>

L_{train} gradient as heatmap.

Blue oval is a contour where L_{train} is a constant value (same color)

Red disc defines uniform weight decay region where $\boldsymbol{\theta}^\top\boldsymbol{\theta} \leq C$.



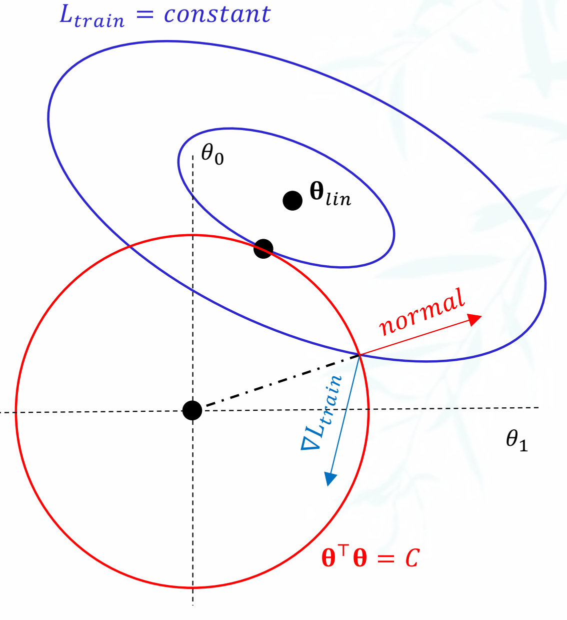
Solving for $\boldsymbol{\theta}_{reg}$

Minimize $L_{train} = \frac{1}{m}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^\top(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})$

Subject to $\boldsymbol{\theta}^\top\boldsymbol{\theta} \leq C$

Observations:

1. Optimal $\boldsymbol{\theta}$ tries to get as 'close' to $\boldsymbol{\theta}_{lin}$ as possible
   Optimal $\boldsymbol{\theta}$ will use full budget and be on the surface $\boldsymbol{\theta}^\top\boldsymbol{\theta} = C$.
2. Surface $\boldsymbol{\theta}^\top\boldsymbol{\theta} = C$, at optimal $\boldsymbol{\theta}$, should be perpendicular to ∇L_{train}.
   Otherwise can move along the surface and decrease L_{train}.
3. Normal to surface $\boldsymbol{\theta}^\top\boldsymbol{\theta} = C$ is the vector $\boldsymbol{\theta}$.
4. Surface is $\perp$ ∇L_{train}; surface is $\perp$ normal.
   ∇L_{train} is parallel to normal (but in the opposite direction).



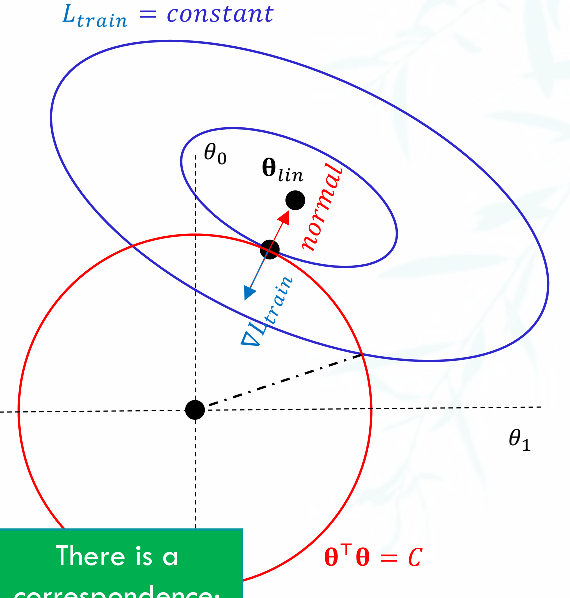
Solving for $\boldsymbol{\theta}_{reg}$

Minimize $L_{train} = \frac{1}{m}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^\top(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})$

Subject to $\boldsymbol{\theta}^\top\boldsymbol{\theta} \leq C$

Observations:

1. Optimal $\boldsymbol{\theta}$ tries to get as 'close' to $\boldsymbol{\theta}_{lin}$ as possible
   Optimal $\boldsymbol{\theta}$ will use full budget and be on the surface $\boldsymbol{\theta}^\top\boldsymbol{\theta} = C$.
2. Surface $\boldsymbol{\theta}^\top\boldsymbol{\theta} = C$, at optimal $\boldsymbol{\theta}$, should be perpendicular to ∇L_{train}.
   Otherwise can move along the surface and decrease L_{train}.
3. Normal to surface $\boldsymbol{\theta}^\top\boldsymbol{\theta} = C$ is the vector $\boldsymbol{\theta}$.
4. Surface is $\perp$ ∇L_{train}; surface is $\perp$ normal.
   ∇L_{train} is parallel to normal (but in the opposite direction).



$$\nabla L_{train}(\boldsymbol{\theta}_{reg}) \propto -\boldsymbol{\theta}_{reg}$$

Constant of proportionality. Chosen for convenience

$$= -2\frac{\lambda}{m}\boldsymbol{\theta}_{reg}$$

There is a correspondence: $C \uparrow \quad \lambda \downarrow$

Comparison with Linear Regression

Unconstrained

min $L_{train} = \frac{1}{m}\sum_{j=1}^{m}\left(\boldsymbol{\theta}^\top \mathbf{x}^{(j)} - y^{(j)}\right)^2$

min $L_{train} = \frac{1}{m}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})$

$\boldsymbol{\theta}_{lin} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}\mathbf{y}$

Constrained:

min $L_{train} = \frac{1}{m}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})$, subject to: $\boldsymbol{\theta}^\top \boldsymbol{\theta} \leq C$

$\boldsymbol{\theta}_{reg} = (\mathbf{X}^\top \mathbf{X} + \boldsymbol{\lambda}\mathbf{I})^{-1}\mathbf{X}^\top \mathbf{y}$

Augmented Error L_{aug}

Unconstrained

min $L_{train} = \frac{1}{m}\sum_{j=1}^{m}\left(\boldsymbol{\theta}^{\top}\mathbf{x}^{(j)} - y^{(j)}\right)^2$

min $L_{train} = \frac{1}{m}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})$

Recall: $\nabla L_{train}\left(\boldsymbol{\theta}_{reg}\right) \propto -\boldsymbol{\theta}_{reg}$
$= -2\frac{\lambda}{m}\boldsymbol{\theta}_{reg}$

$\boldsymbol{\theta}_{lin} = (\mathbf{X}^{\top}\mathbf{X})^{-1}\mathbf{X}\mathbf{y}$

Constrained:

min $L_{train} = \frac{1}{m}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})$, subject to: $\boldsymbol{\theta}^{\top}\boldsymbol{\theta} \leq C$

$\equiv$

min $L_{aug}(\boldsymbol{\theta}) = L_{train}(\boldsymbol{\theta}) + \frac{\lambda}{m}\boldsymbol{\theta}^{\top}\boldsymbol{\theta}$
$= \frac{1}{m}\left((\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^{\top}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y}) + \lambda\boldsymbol{\theta}^{\top}\boldsymbol{\theta}\right)$

Compare versus implicit constraint

Take derivatives:

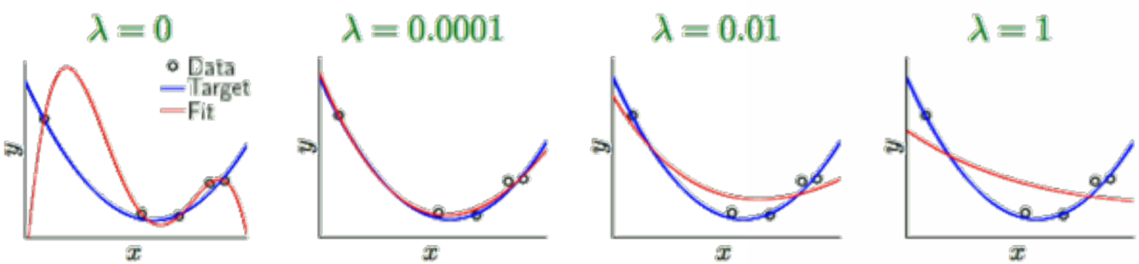
Set $\nabla L_{aug}(\boldsymbol{\theta}) = \overleftarrow{\mathbf{0}} \Longrightarrow$
$= \mathbf{X}^{\top}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y}) + \boldsymbol{\lambda}\boldsymbol{\theta}$
$= (\mathbf{X}^{\top}\mathbf{X} + \boldsymbol{\lambda}\mathbf{I})\boldsymbol{\theta} - \mathbf{X}^{\top}\mathbf{y}$

$\boldsymbol{\theta}_{reg} = (\mathbf{X}^{\top}\mathbf{X} + \boldsymbol{\lambda}\mathbf{I})^{-1}\mathbf{X}^{\top}\mathbf{y}$

Still 1 step learning! λ determines the amount of regularization.

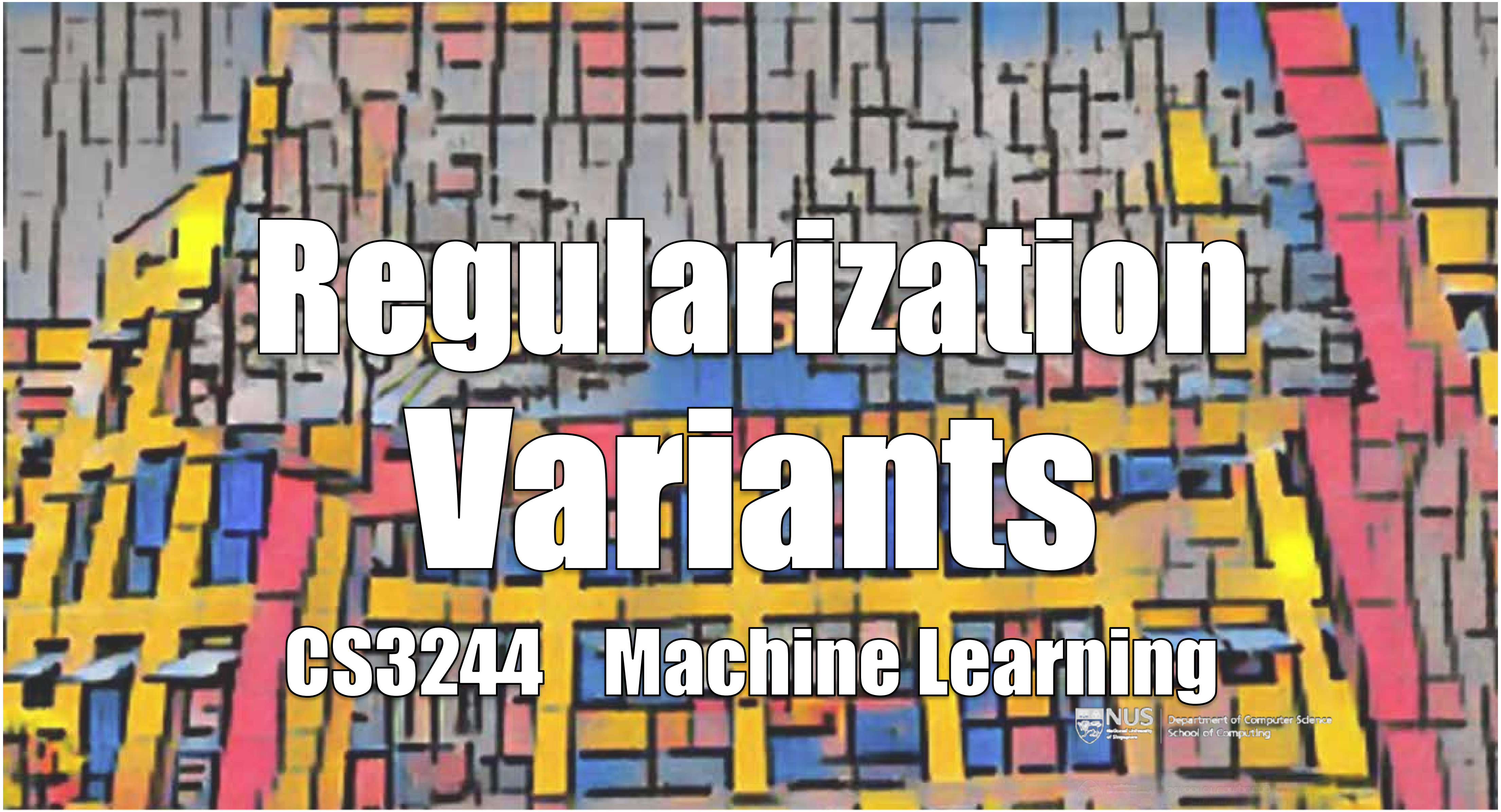
Take the right λ amount of medicine

Minimizing $L_{train}(\boldsymbol{\theta}) + \frac{\lambda}{m}\boldsymbol{\theta}^\top\boldsymbol{\theta}$ for different λ's:



overfitting $\rightarrow$ $C \rightarrow 0$ $\lambda \rightarrow \infty$ $\rightarrow$ underfitting

> Q1: What happens $\boldsymbol{\theta}$ to in the limit as $\lambda \rightarrow \infty$?



Regularization Variants

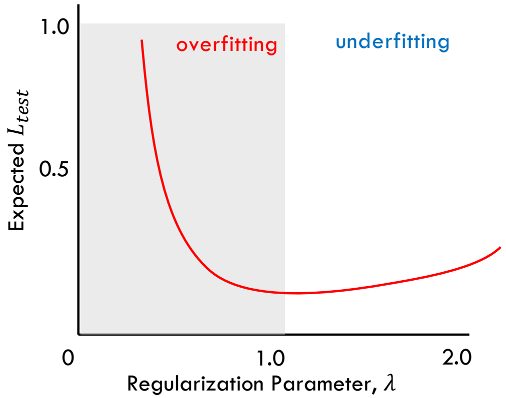
CS3244 Machine Learning

NUS
Department of Computer Science
School of Computing

Regularizer variations $\Omega(h)$

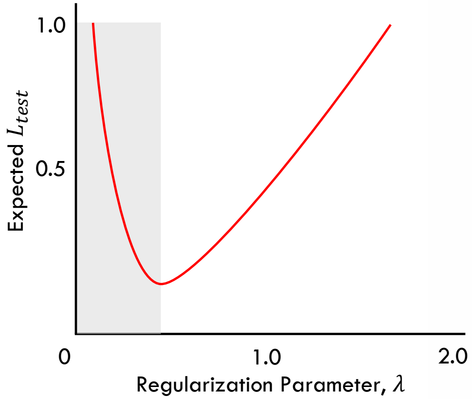
Why is it called "weight decay"? We'll see in neural networks.

Uniform weight decay



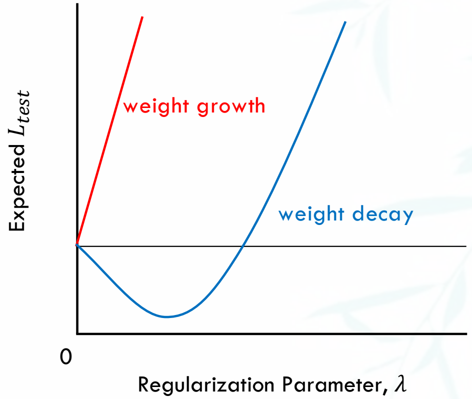
$$\sum_{q=0}^{Q} \theta_q^2$$

Low Order Fit



$$\sum_{q=0}^{Q} q\theta_q^2$$

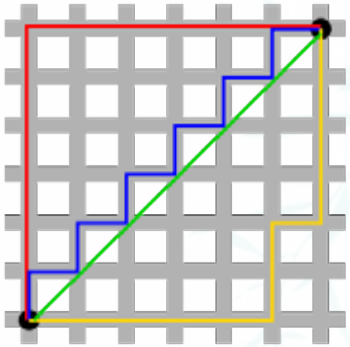
Weight growth



$$\sum_{q=0}^{Q} \frac{1}{\theta_q^2}$$

Geometry of ℓ^p norms

$$||x||_p \equiv \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}$$



ℓ^1 = Manhattan (Taxicab) distance

$\sum_{i=1}^{n} |x_i|$

ℓ^2 = Euclidean distance (weight decay)

$\sqrt[2]{\sum_{i=1}^{n} |x_i|^2}$

Diagram Credit: Psychonaut via Wikipedia, public domain

ℓ^p ball visualized

As the value of p decreases, the size of the corresponding space also decreases.

In 3 equally weighted dimensions (e.g., x_1, x_2, x_3):

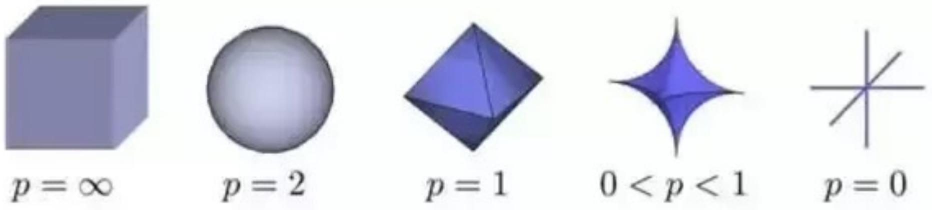
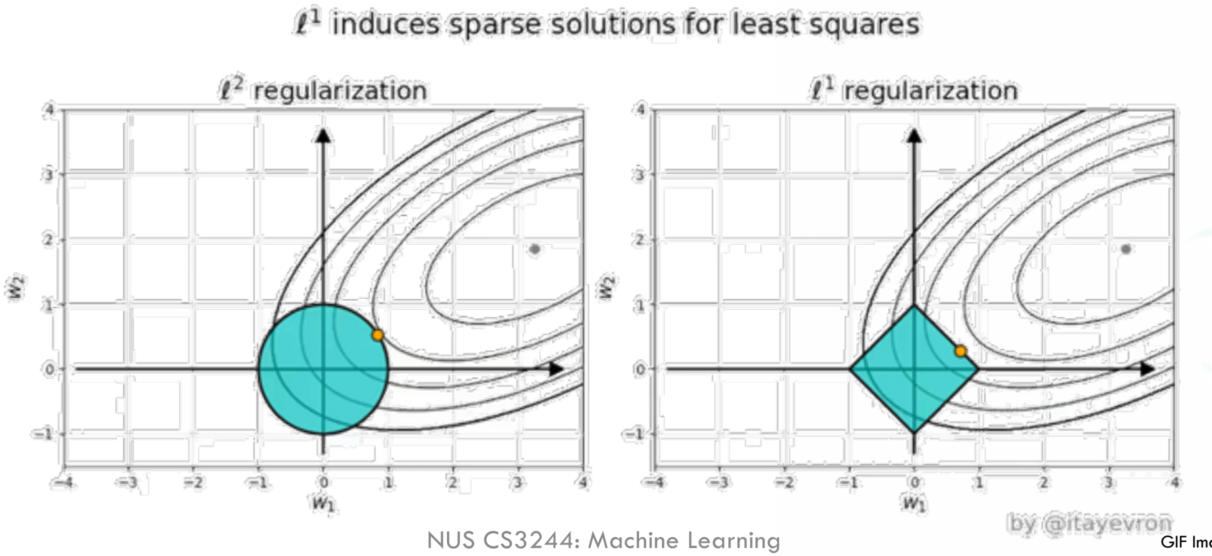


Diagram Credit: Manjari Narayan via Quora.

Properties from geometry

ℓ^1 encourages sparse solutions; akin to feature selection.

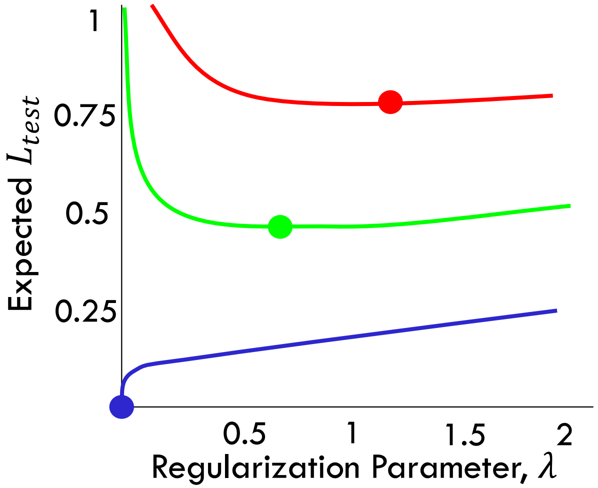
ℓ^2 can be used for homogenous data.



GIF Image from [Itay Evron]

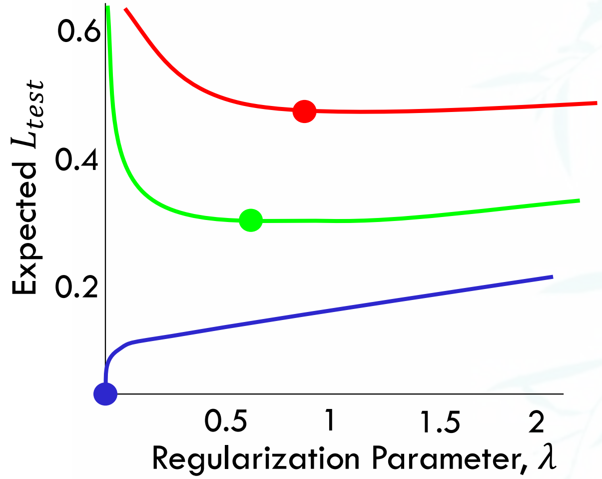
Getting the right dose

Recall the bias-variance experiment varying σ^2 (noise) and Q_f (target complexity) :



Stochastic Noise

(high frequency)



Deterministic Noise (Bias)

(also non-smooth w.r.t. to $\mathcal{H}$)

The perfect regularizer $\Omega(h)$

Constraint in the 'direction' of the target function

But we don't know f: circular argument!

Guiding principle:

Direction of a smoother or "simpler" hypothesis

Smoother = impairs our ability to fit (high-frequency) noise

Sacrifice a little bias for large improvement on variance

Chose a bad Ω?

We still can tune λ! $\Longrightarrow$ validation, up next

Regularization – Summary

Give up modeling a subset of of $\mathcal{H}$ to lower variance error.

$L_{aug}(\boldsymbol{\theta}) = L_{train}(\boldsymbol{\theta}) + \frac{\lambda}{m}\Omega(h)$

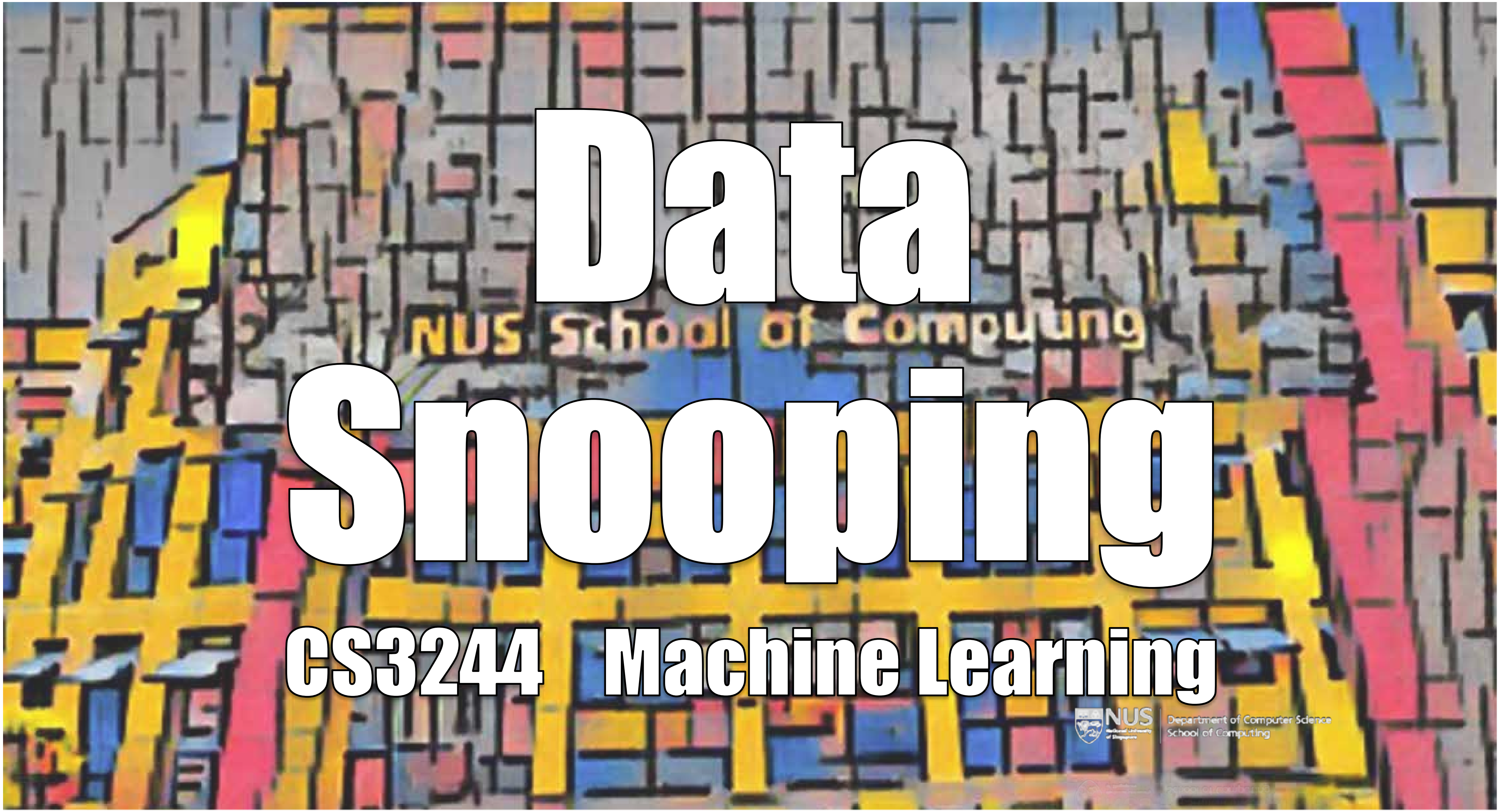
λ is the dose of regularization:

the higher the dose, the smaller the budget C.

Choosing a regularizer $\Omega(h)$:

Weight decay is common $\sum_{q=0}^{Q} \theta_q^2 \equiv \boldsymbol{\theta}^\top \boldsymbol{\theta}$ (ℓ^2 regularization)

Choose λ by validation …


Data Snooping
NUS School of Computing
CS3244 Machine Learning
NUS
Department of Computer Science
School of Computing

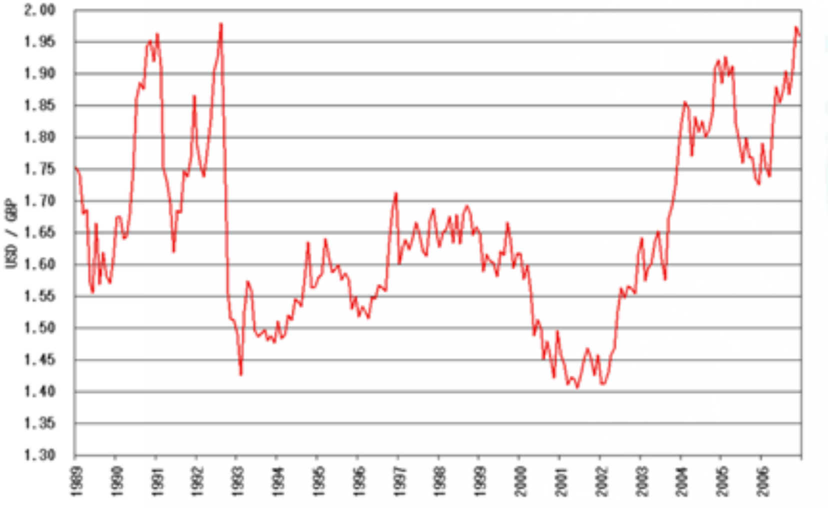
Data Snooping

Predict USD versus GBP

Normalize data, split randomly: $\mathbf{X}_{train}, \mathbf{X}_{test}$

Train only on $\mathbf{X}_{train}$,
Test h_θ on $\mathbf{X}_{test}$

Got great performance!
Let's invest!



$\Delta r_{-20}, \Delta r_{-19}, \ldots, \Delta r_{-1} \rightarrow \Delta r_0$

Chart credits: Monaneko @ Wikimedia Commons

Data Snooping

In Zoom breakout or physical subgroups,

(5 mins): Answer why we lost our money 💰💸💸

Ask one member to write it to the #general thread. Upvote others that you like.

$\Delta r_{-20}, \Delta r_{-19}, \ldots, \Delta r_{-1} \rightarrow \Delta r_0$

Chart credits: Monaneko @ Wikimedia Commons

Data Snooping

Lost our $$$. Why?

If a data set has affected any step in the learning process, its ability to assess the outcome has been compromised.

Most common trap for practitioners – many ways to slip.

Chart credits: Monaneko @ Wikimedia Commons

Validation

CS3244 Machine Learning

Two Cures

In one form or another, $L_{test}(h) = L_{train}(h) +$ overfit penalty

1. **Regularization**: Restrain the model
$$L_{test}(h) = L_{train}(h) + \text{overfit penalty}$$

2. **Validation**: Reality check by peeking (at the bottom line)
$$L_{test}(h) = L_{train}(h) + \text{overfit penalty}$$

Validation estimates this quantity

Analyzing the estimated loss

On a test point (x, y), the cost $l(h_\theta(x), y)$ is:

Squared error: $(h_\theta(x) - y)^2$

Binary error: $[[h_\theta(x) \neq y]]$

$$\mathbb{E}[l(h_\theta(x), y)] = L_{test}(h_\theta)$$

$$Var[l(h_\theta(x), y)] = \sigma^2$$

From a point to a set

On a validation set $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(K)}, y^{(K)})$,
the cost is $L_{val}(h) = \frac{1}{K}\sum_{k=1}^{K} l\left(h(\mathbf{x}^{(k)}), y^{(k)}\right)$

$$\mathbb{E}[L_{val}(h)] = \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\left[l\left(h(\mathbf{x}^{(k)}), y^{(k)}\right)\right] = L_{test}(h)$$

$$Var[L_{val}(h)] = \frac{1}{K^2}\sum_{k} Var\left[l\left(h(\mathbf{x}^{(k)}), y^{(k)}\right)\right] = \frac{\sigma^2}{K}$$

Our K points are i.i.d.

$$L_{val}(h) = L_{test}(h) \pm O\left(\frac{1}{\sqrt{K}}\right)$$

Standard Deviation

K is taken out of m

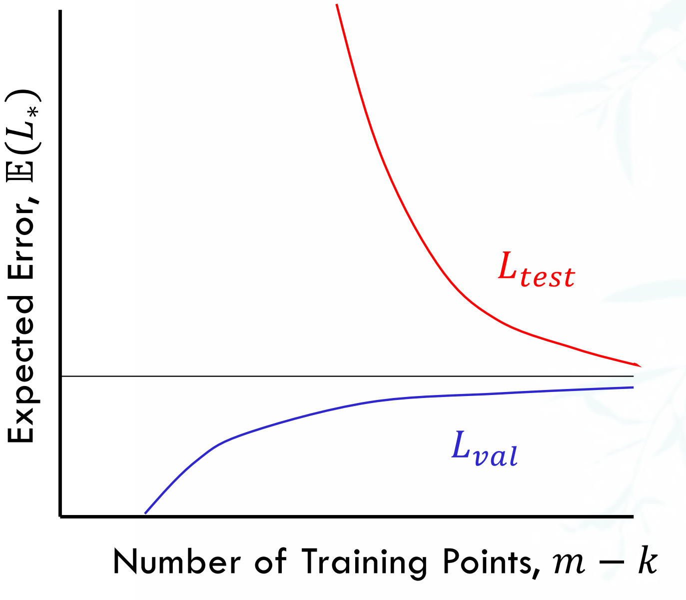
Given the data set $\mathcal{D}$, separate:

1. K points for validation
2. $m - K$ points for training

$O\left(\frac{1}{\sqrt{K}}\right)$:

Small K = bad estimate

Large K = ?



K is put back into m

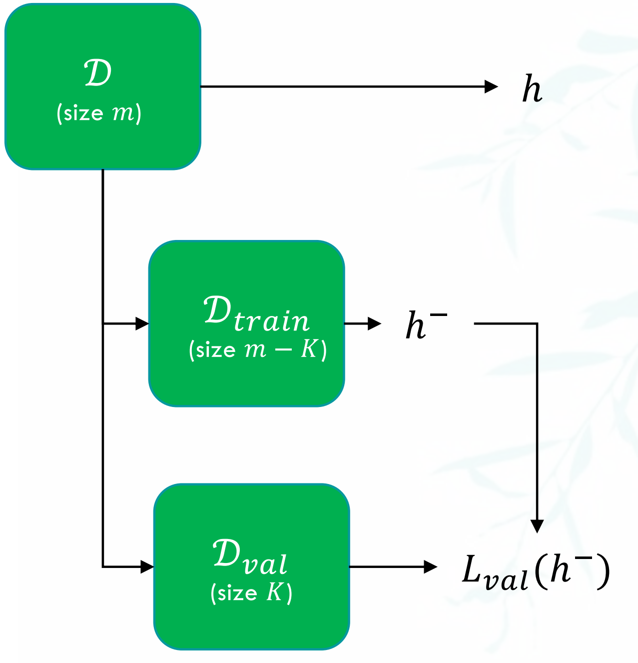
1. Train on $\mathcal{D}_{train}$ to yield h^-
2. Test h^- on $\mathcal{D}_{val}$ to yield L_{val}
3. Use cost L_{val} to estimate $L_{test}(h^-)$
4. Use h (not h^-!) in the end

Large K?

h^- trained on too few examples.

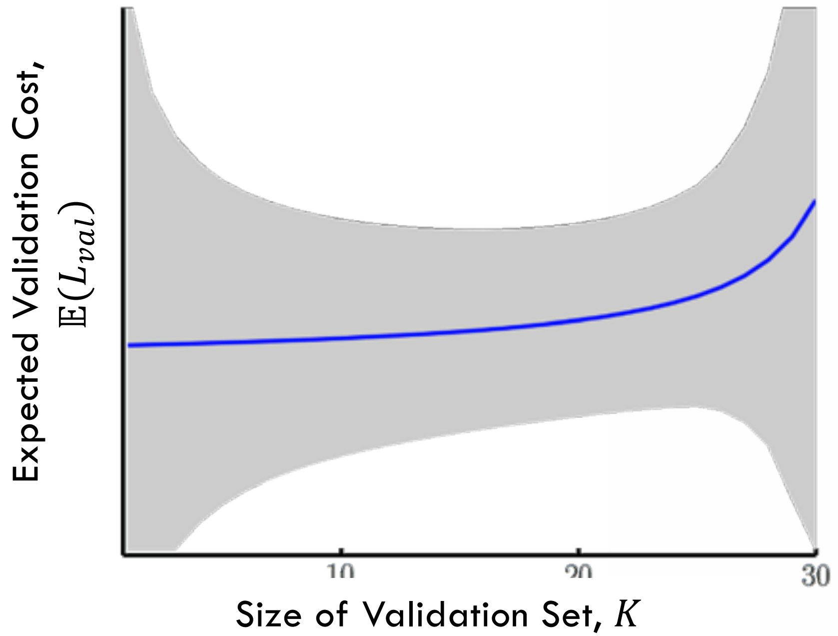
Leads to bad h^-, poor estimate.

Rule of Thumb: $K = \frac{m}{5}$



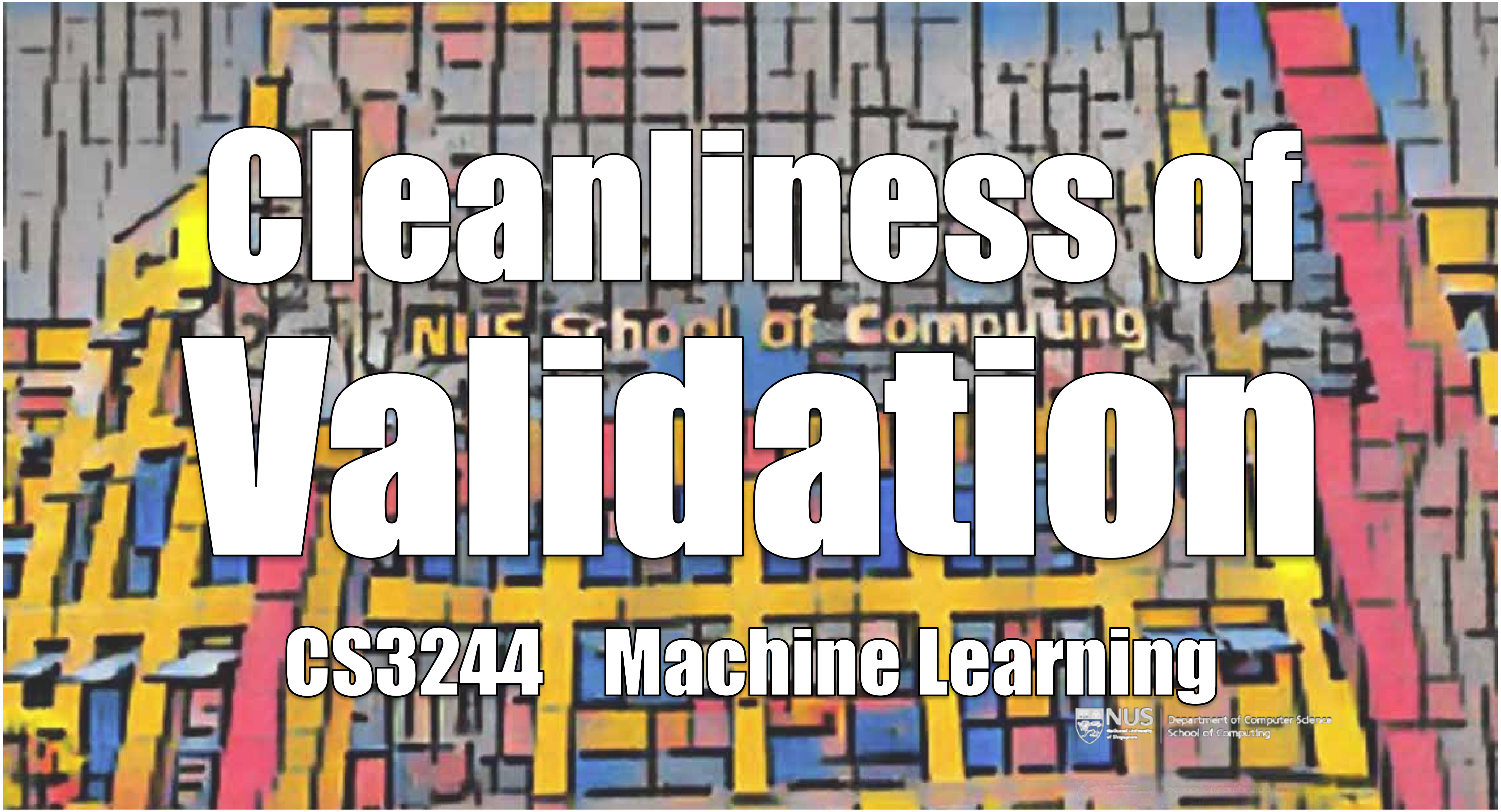
Expected Validation Error for $\mathcal{H}_2$

With $m = 40$, and noise level = 0.4



Cleanliness of Validation

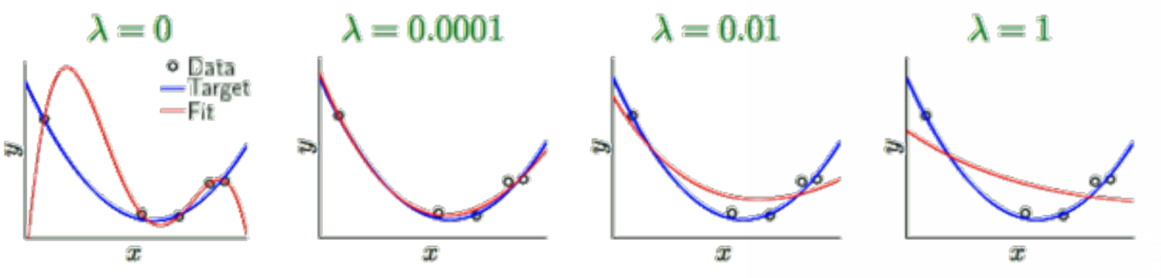
CS3244 Machine Learning



NUS National University of Singapore | Department of Computer Science
School of Computing

Why 'validation'?

Because $\mathcal{D}_{val}$ is used to make learning choices,
e.g. choosing the level of regularization to minimize L_{val}



If an estimate L_{test} affects learning:

The set is no longer a test set, it becomes a validation set!

What's the difference?

We know that the test set is unbiased.
But what about the validation set?

Your Turn: Does the validation set have an optimistic or pessimistic bias?

Two hypotheses h_a and h_b with $L_{test}(h_a) = L_{test}(h_b) = 0.5$

Error estimates l_a and l_b uniform on $[0,1]$

We pick $h \in \{h_a, h_b\}$ by virtue of its $l = \min(l_a, l_b)$.

What then, is the value of $\mathbb{E}(l)$?

[ ] leading to [ ] bias.

What's the difference?

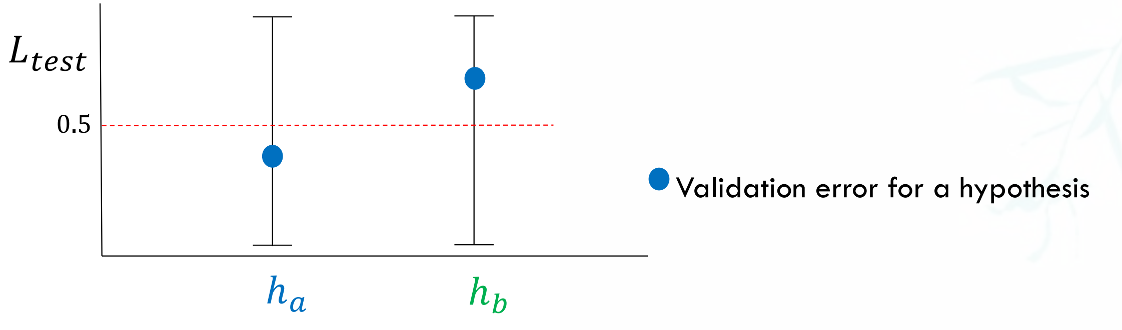
Two hypotheses h_a and h_b with $L_{test}(h_a) = L_{test}(h_b) = 0.5$

Error estimates l_a and l_b uniform on $[0,1]$

We pick $h \in \{h_a, h_b\}$ by virtue of its $l = \min(l_a, l_b)$.

What then, is the value of $\mathbb{E}(l)$?

$\mathbb{E}(l) < 0.5$, leading to an optimistic bias.



Hypothesis Selection: Using $\mathcal{D}_{val}$ more than once

Θ models $\mathcal{H}_1, \dots, \mathcal{H}_\Theta$

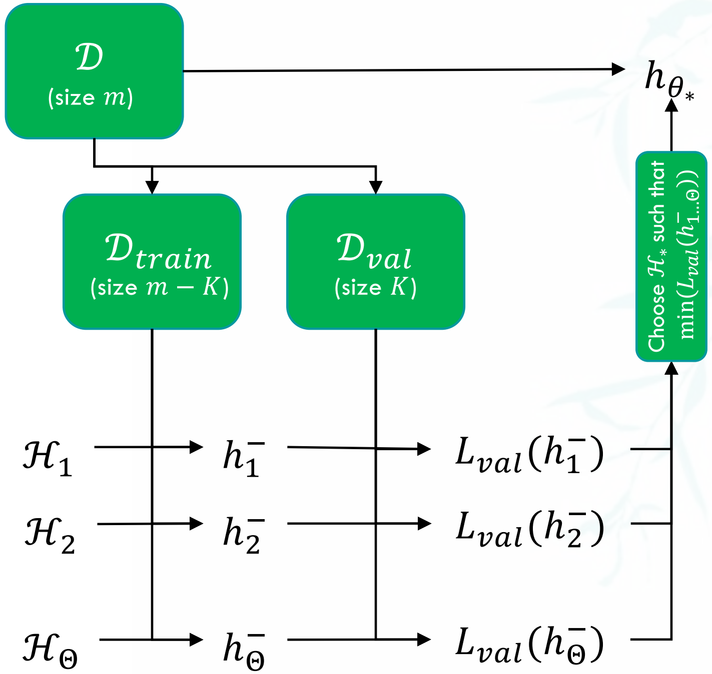
Use $\mathcal{D}_{train}$ to learn h_θ^- for each model

Evaluate h_θ^- using $\mathcal{D}_{val}$:

$$L_\theta = L_{val}(h_\theta^-); \;\; \theta = 1, \dots, \Theta$$

<u>Pick</u> model $\theta = \theta_*$ with smallest L_θ

Uh oh!
Bias!



Data contamination

Error estimates: L_{train}, L_{test}, L_{val}

Contamination: optimistic (deceptive) bias in estimating L_{test}

- Training set: totally contaminated
- Validation set: slightly contaminated
- Test set: totally 'clean'

Time Out: Check your understanding

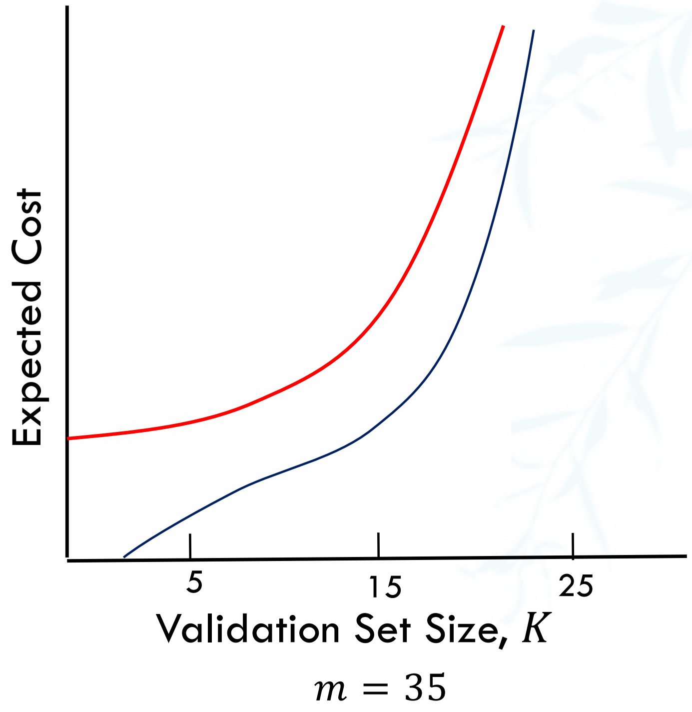
We selected the model $\mathcal{H}_{\theta^*}$ using $\mathcal{D}_{val}$. We can think of validation as training among the set of Θ models.

$L_{val}(h_{\bar{\theta}^*})$ is a biased estimate of $L_{test}(h_{\bar{\theta}^*})$

Your Turn:

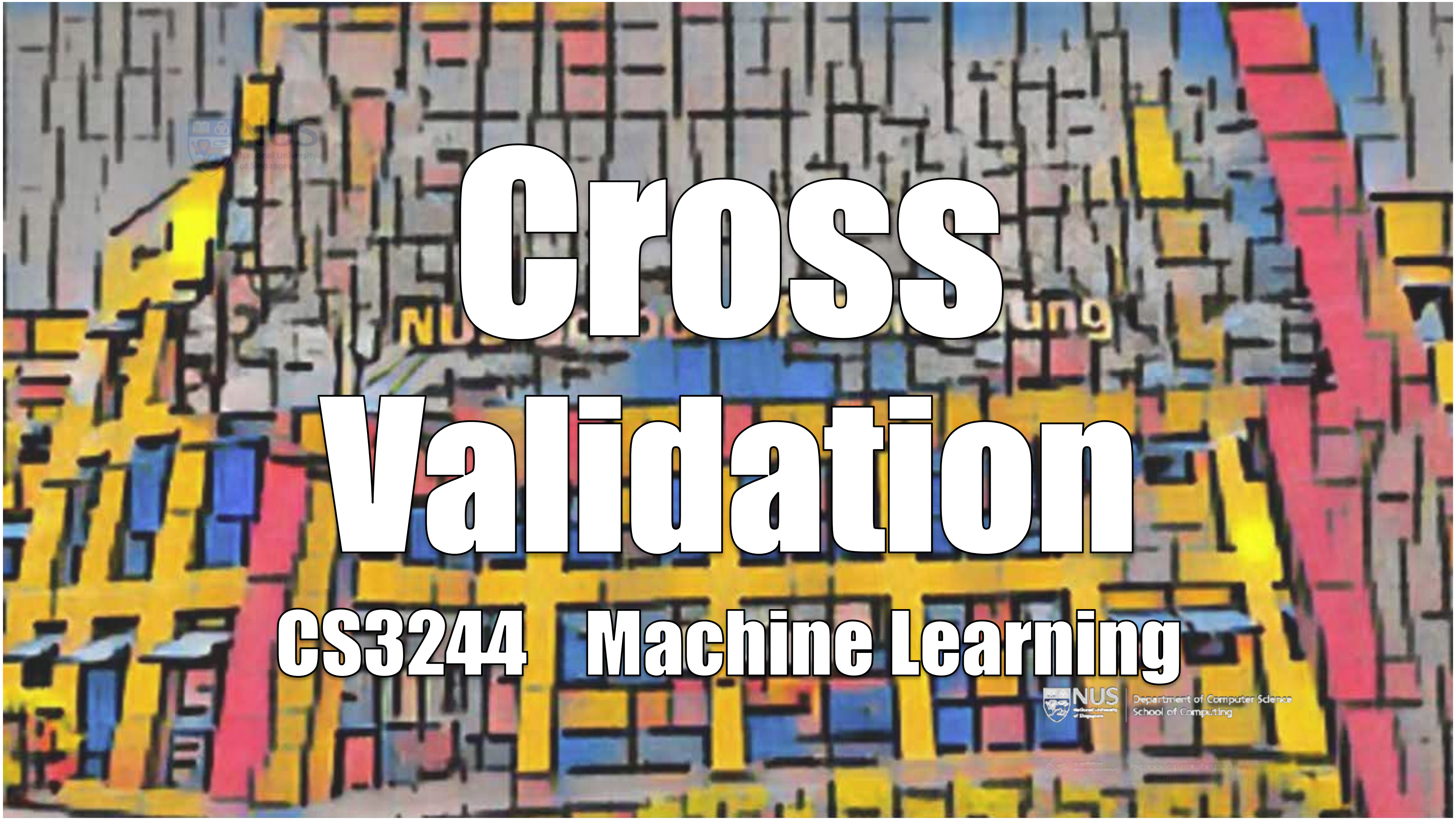
One curve is L_{val}, and the other L_{test}

1. Which curve is L_{val}? 🔴 or 🔵
2. Why are the curves going up?
3. Why do the curves get closer together?



Cross Validation

CS3244 Machine Learning



The dilemma about K

Validation relies on the following chain of reasoning:

True when K is big

$$L_{test}(h) \approx L_{test}(h^-) \approx L_{val}(h^-)$$

True when K is small

Can we have both K being both big and small?

Yes, we can!

Leave out out cross validation

$m - 1$ points for training and 1 point for validation
(Sounds familiar? It was at the beginning of the pre validation lecture)

$\mathcal{D}_{cv} = \left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \cancel{\left(\mathbf{x}^{(cv)}, y^{(cv)}\right)}, \ldots, (\mathbf{x}^{(m)}, y^{(m)})$ validation

Final hypothesis learned from $\mathcal{D}_{cv}$ is h_{cv}^{-}.

$$l_{cv} = l_{val}(h_{cv}^{-}) = l\left(h_{cv}^{-}\left(\mathbf{x}^{(cv)}\right), y^{(cv)}\right)$$

Caveat: Hypothesis learned will be highly correlated.
As most points are identical: The 1st hypothesis will use points 2, 3, …, m and the 2nd will use 1, 3, … , m.

Leave out out cross validation

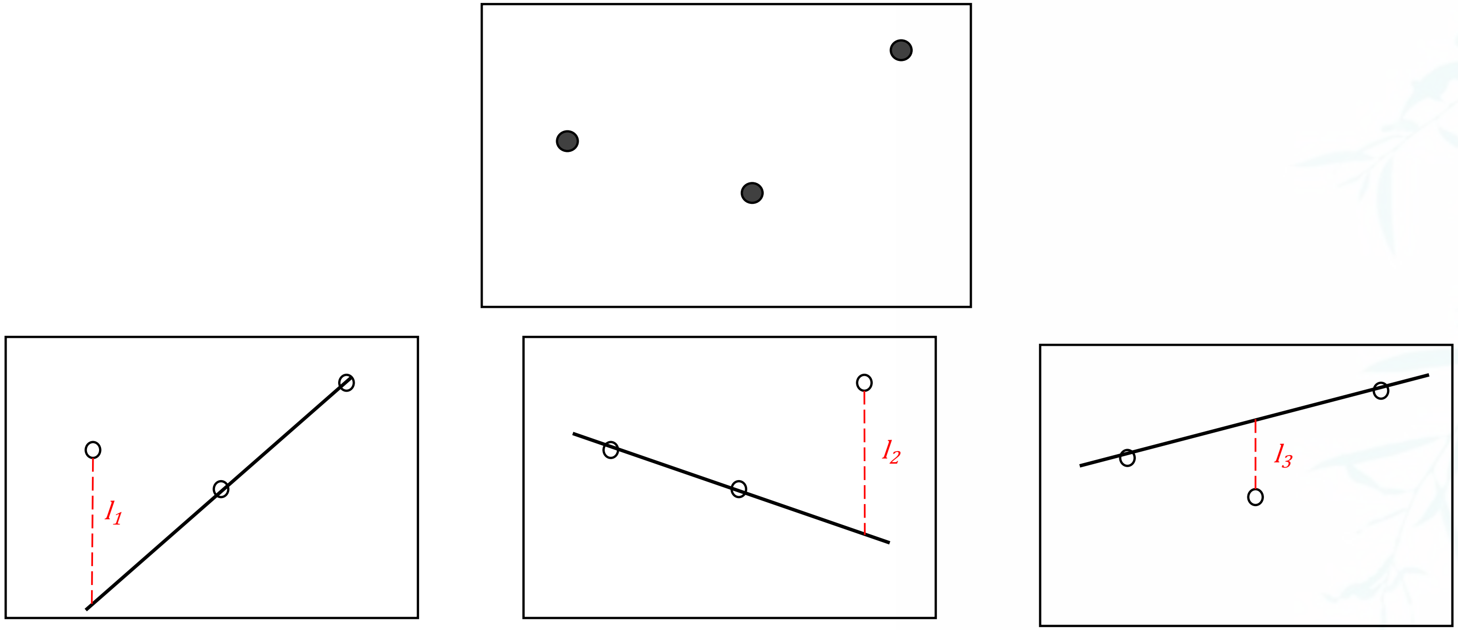
Final hypothesis learned from $\mathcal{D}_{cv}$ is h_{cv}^{-}.

$$l_{cv} = l_{val}(h_{cv}^{-}) = l\left(h_{cv}^{-}\left(\mathbf{x}^{(cv)}\right), y^{(cv)}\right)$$

Cross validation cost: $L_{loocv} = \frac{1}{m}\sum_{cv=1}^{m} l_{cv}$.

(Almost) using m examples for K.

Illustration of cross validation

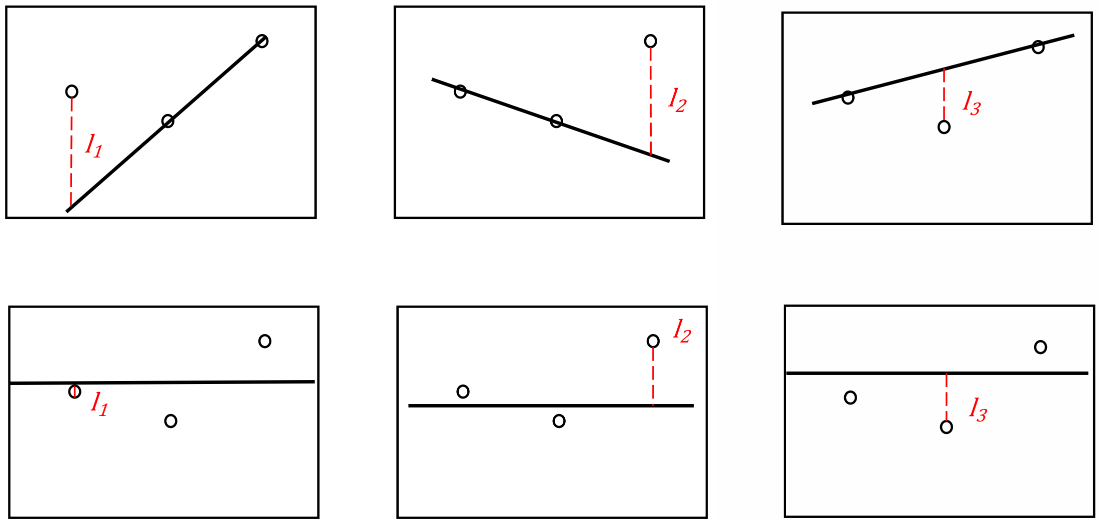


$$L_{loocv} = \frac{1}{3}(l_1 + l_2 + l_3)$$

Model selection using CV

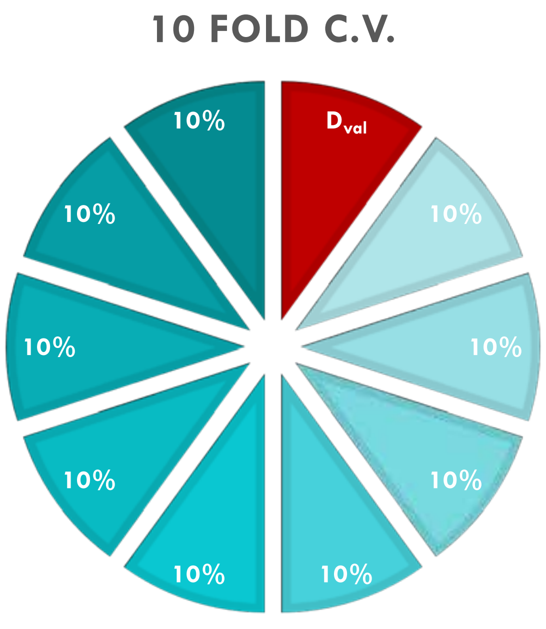
Linear

Constant



L_{cv} empirically shows that the constant model is a better fit for this dataset

Have your cake and eat it too: K fold cross validation



LOOCV can be very expensive for large datasets. Why?

Instead, use K fold cross validation: i.e., K training sessions on $\frac{m}{K}$ points each.

Recommend: 10-fold CV

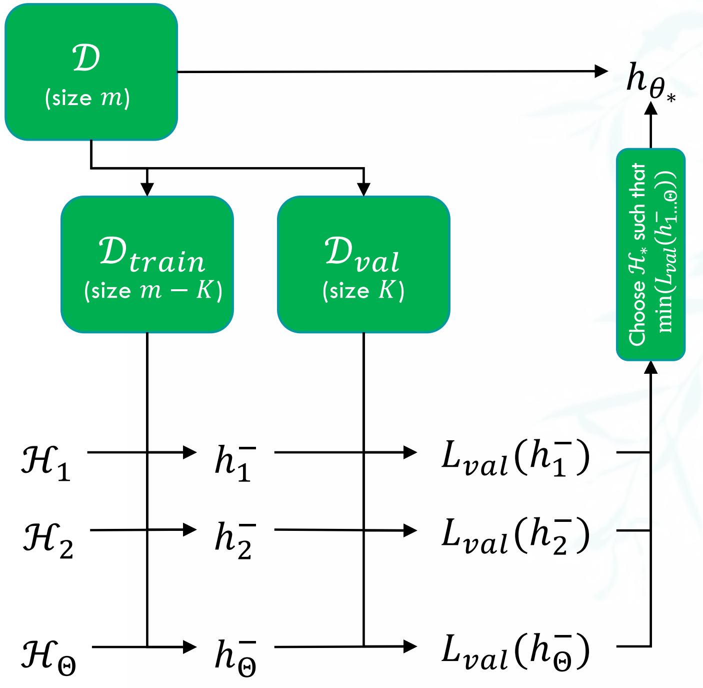
Cross Validation – Summary

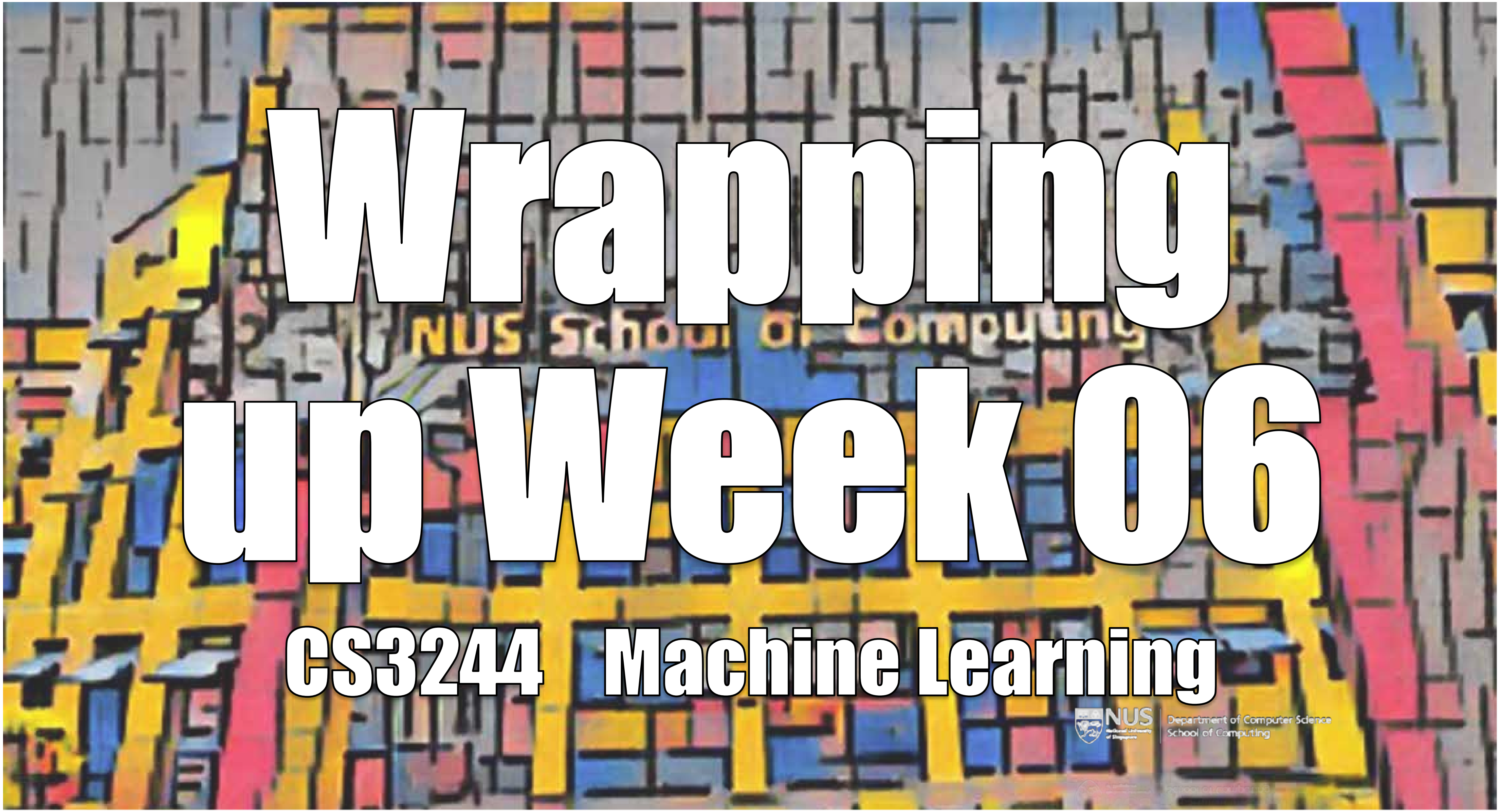
Estimate L_{val} multiple times

Breaks independence assumption: performance is correlated.

Introduces the factor of efficiency as a tradeoff.

To think about: How else can we produce estimates of L_{test}?




Wrapping up Week 06
NUS School of Computing
CS3244 Machine Learning
NUS
Department of Computer Science
School of Computing

What did we learn this week?

Understand **Regularization** as a means of restraining the model.

Choose appropriate doses of regularization for a model.

Understand and execute **Validation**, as a reality check by peeking (at the bottom line).

Understand the different forms extending validation to encompass additional estimation.

Understand how validation and regularization complement each other and their roles in affecting learning.

Outlook for next week

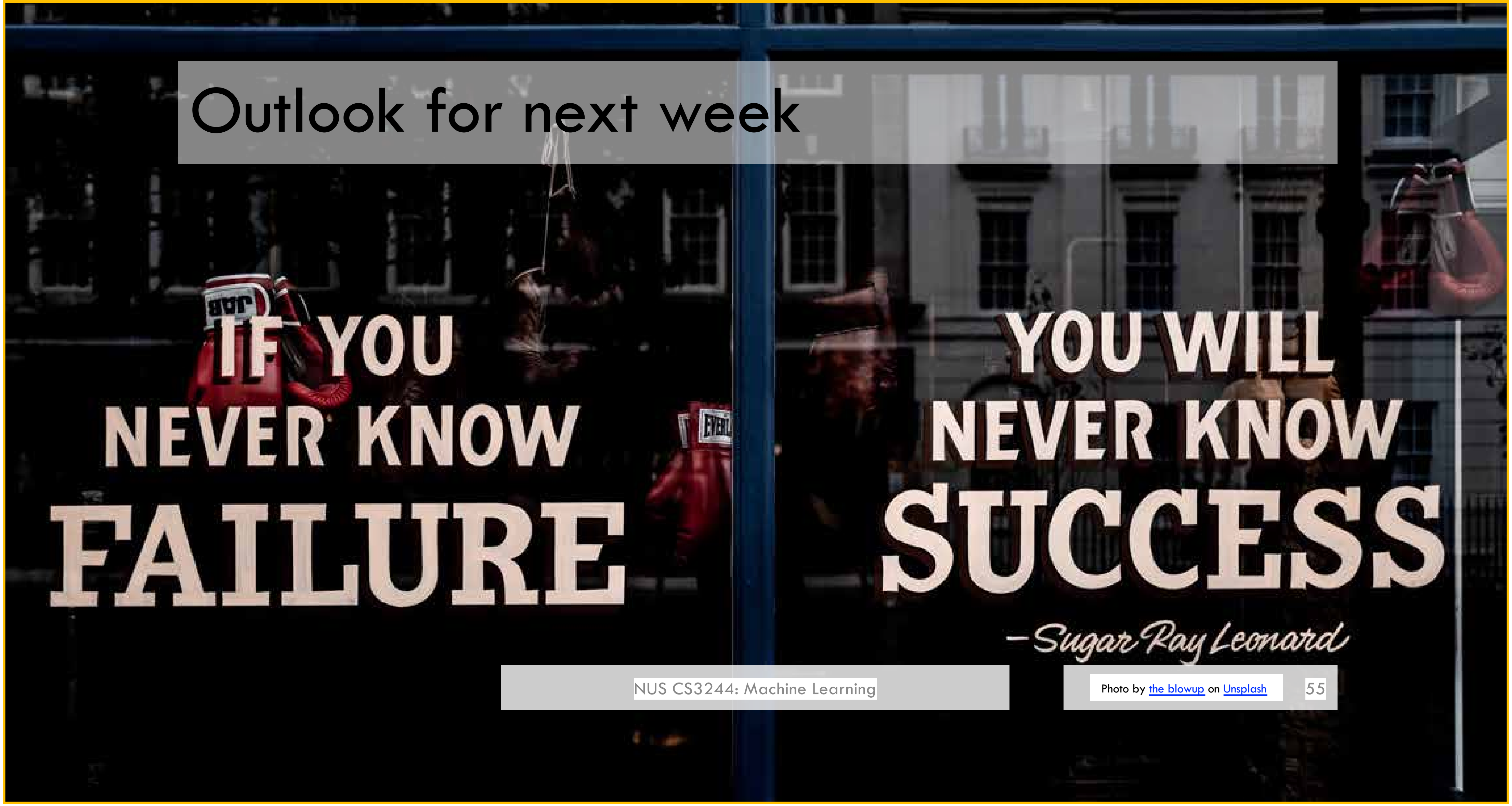


Photo by [the blowup](#) on [Unsplash](#)

Assigned Task (due before next Mon)

Take a break, you deserve it!