Learning From Data
Lecture 11
Overfitting

What is Overfitting
When does Overfitting Occur
Stochastic and Deterministic Noise

M. Magdon-Ismail
CSCI 4100/6100
RECAP: Nonlinear Transforms

1. Original data
   \( x_n \in \mathcal{X} \)

2. Transform the data
   \( z_n = \Phi(x_n) \in \mathcal{Z} \)

3. Separate data in \( \mathcal{Z} \)-space
   \( \tilde{g}(z) = \text{sign}(\tilde{w}^Tz) \)

4. Classify in \( \mathcal{X} \)-space
   \( g(x) = \tilde{g}(\Phi(x)) = \text{sign}(\tilde{w}^T\Phi(x)) \)

\( \mathcal{X} \)-space is \( \mathbb{R}^d \)
\( \mathcal{Z} \)-space is \( \mathbb{R}^{\tilde{d}} \)

\( x = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \)
\( z = \Phi(x) = \begin{bmatrix} 1 \\ \Phi_1(x) \\ \vdots \\ \Phi_{\tilde{d}}(x) \end{bmatrix} = \begin{bmatrix} 1 \\ z_1 \\ \vdots \\ z_{\tilde{d}} \end{bmatrix} \)

\( x_1, x_2, \ldots, x_N \)
\( z_1, z_2, \ldots, z_N \)
\( y_1, y_2, \ldots, y_N \)
\( y_1, y_2, \ldots, y_N \)

no weights

\( \tilde{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{\tilde{d}} \end{bmatrix} \)

\( d_{vc} = d + 1 \)

Overfitting: 2 / 25
RECAP: Digits Data “1” Versus “All”

**Linear model**
\[
E_{\text{in}} = 2.13\%
\]
\[
E_{\text{out}} = 2.38\%
\]

**3rd order polynomial model**
\[
E_{\text{in}} = 1.75\%
\]
\[
E_{\text{out}} = 1.87\%
\]
Superstitions – Myth or Reality?

- **Paraskevedekatriaphobia** – fear of Friday the 13th.
  – Are future Friday the 13ths really more dangerous?

- **OCD** [medical journal, citation lost, can you find it?]
  
  the subjects performs an action which leads to a good outcome and thereby generalizes it as cause and effect: the action will always give good results. Having *overfit* the data, the subject compulsively engages in that activity.

Humans are **overfitting machines**, very good at “finding coincidences”.
An Illustration of Overfitting on a Simple Example

Quadratic $f$

5 data points

A little noise (measurement error)

5 data points $\rightarrow$ 4th order polynomial fit
An Illustration of Overfitting on a Simple Example

Quadratic $f$

5 data points

A little noise (measurement error)

5 data points $\rightarrow$ 4th order polynomial fit

Classic overfitting: simple target with excessively complex $\mathcal{H}$.

$E_{\text{in}} \approx 0; \ E_{\text{out}} \gg 0$

The noise did us in. (why?)
What is Overfitting?

Fitting the data more than is warranted
Overfitting is Not Just Bad Generalization

VC Analysis:
Covers bad generalization but with lots of slack – the VC bound is loose
Overfitting is Not Just Bad Generalization

Overfitting:

Going for lower and lower $E_{in}$ results in higher and higher $E_{out}$
Case Study: 2nd vs 10th Order Polynomial Fit

\[ f \] with noise. 50th order \[ f \] with no noise.

\[ H_2 \]: 2nd order polynomial fit
\[ H_{10} \]: 10th order polynomial fit

\[ x \mapsto (1, x, x^2, \cdots) \].

Which model do you pick for which problem and why?

\[ H_2 \text{ versus } H_{10} \]

10th order \( f \) with noise.

50th order \( f \) with no noise.
Case Study: 2nd vs 10th Order Polynomial Fit

10th order $f$ with noise.

50th order $f$ with no noise.

$H_2$: 2nd order polynomial fit

$H_{10}$: 10th order polynomial fit

← special case of linear models with feature transform $x \mapsto (1, x, x^2, \cdots)$.

Which model do you pick for which problem and why?
Case Study: 2nd vs 10th Order Polynomial Fit

Go figure:

Simpler $\mathcal{H}$ is better even for the more complex target with no noise.
Is there Really “No Noise” with the Complex $f$?

Simple $f$ with noise.

Complex $f$ with no noise.
Is there Really “No Noise” with the Complex $f$?

Simple $f$ with noise.  

Complex $f$ with no noise.

$\mathcal{H}$ should match *quantity and quality of data*, not $f$
When is \( \mathcal{H}_2 \) Better than \( \mathcal{H}_{10} \)?

Overfitting:

\[
E_{out}(\mathcal{H}_{10}) > E_{out}(\mathcal{H}_2)
\]
Overfit Measure: $E_{\text{out}}(\mathcal{H}_{10}) - E_{\text{out}}(\mathcal{H}_{2})$
Overfit Measure: $E_{\text{out}}(\mathcal{H}_1) - E_{\text{out}}(\mathcal{H}_2)$

Number of Data Points, $N$

Noise Level, $\sigma^2$

Target Complexity, $Q_f$

Number of data points $\uparrow$ Overfitting $\downarrow$

Noise $\uparrow$ Overfitting $\uparrow$

Target complexity $\uparrow$ Overfitting $\uparrow$

© Creator: Malik Magdon-Ismail
That part of $y$ we cannot model

it has two sources ...
We would like to learn from $\circ$:
\[ y_n = f(x_n) \]

Unfortunately, we only observe $\circ$:
\[ y_n = f(x_n) + \text{‘stochastic noise’} \]

Stochastic Noise: fluctuations/measurement errors we cannot model.
We would like to learn from $\circ$:
\[ y_n = h^*(x_n) \]

Unfortunately, we only observe $\circ$:
\[ y_n = f(x_n) = h^*(x_n) + \text{‘deterministic noise’} \]
\[ H \text{ cannot model this} \]

**Deterministic Noise:** the part of $f$ we cannot model.
Stochastic & Deterministic Noise Hurt Learning

Stochastic Noise

\[ y = f(x) + \text{stoch. noise} \]

**source:** random measurement errors
re-measure \( y_n \)
stochastic noise changes.
change \( \mathcal{H} \)
stochastic noise the same.

Deterministic Noise

\[ y = h^*(x) + \text{det. noise} \]

**source:** learner’s \( \mathcal{H} \) cannot model \( f \)
re-measure \( y_n \)
deterministic noise the same.
change \( \mathcal{H} \)
deterministic noise changes.

We have single \( \mathcal{D} \) and fixed \( \mathcal{H} \) so we cannot distinguish
Noise and the Bias-Variance Decomposition

\[ y = f(x) + \epsilon \]

\[ \uparrow \]

\[ \text{measurement error} \]

\[ \mathbb{E}[E_{out}(x)] = \mathbb{E}_{D,\epsilon}[(g(x) - f(x) - \epsilon)^2] \]

\[ = \mathbb{E}_{D,\epsilon}[(g(x) - f(x))^2 + 2(g(x) - f(x))\epsilon + \epsilon^2] \]

\[ \downarrow \quad \downarrow \quad \downarrow \]

\[ \text{bias + var} \quad 0 \quad \sigma^2 \]
Noise and the Bias-Variance Decomposition

\[ y = f(x) + \epsilon \]

measurement error

\[ \mathbb{E}[E_{out}(x)] = \sigma^2 + \text{bias} + \text{var} \]

stochastic noise
deterministic noise
indirect impact of noise

Overfitting: 23/25
Noise is the Culprit

Overfitting is the disease

Noise is the cause

Learning is led astray by fitting the noise more than the signal

Cures

Regularization: Putting on the brakes.

Validation: A reality check from peeking at $E_{\text{out}}$ (the bottom line).
Regularization

no regularization  regularization!

\[ y \]

Data
Target
Fit

\[ x \]

Overfitting: 25/25
Regularization

no regularization

regularization!

Data
Target
Fit