CSE 417T
Introduction to Machine Learning

Lecture 10
Instructor: Chien-Ju (CJ) Ho
Logistics

• Homework 2: due on **Mar 8, Monday**

• Exam 1: **Mar 23 (Tuesday)**

• No class next Tuesday (Wellness day)
Recap
Linear Models

- $H$ contains hypothesis $h(\vec{x})$ as some function of $\vec{w}^T \vec{x}$

- Algorithm:
  - Focus on $g = \arg\min_{h \in H} E_{in}(h)$
  - Gradient descent is one of the common optimization algorithms
Gradient Descent

• Gradient descent algorithm
  • Initialize $\vec{w}(0)$
  • For $t = 0, ...$
    • $\vec{w}(t + 1) \leftarrow \vec{w}(t) - \eta \nabla_{\vec{w}} E_{in}(\vec{w}(t))$
    • Terminate if the stop conditions are met
  • Return the final weights

• Stochastic gradient decent
  • Replace the update step:
    • Randomly choose $n$ from \{1, ..., $N$\}
    • $\vec{w}(t + 1) \leftarrow \vec{w}(t) - \eta \nabla_{\vec{w}} e_n(\vec{w}(t))$

Works for functions where gradient exists everywhere
Nonlinear Transformation

\[ \hat{z} = \Phi(\hat{x}) \]

\[ g(z)(\hat{z}) = \text{sign}(\overrightarrow{w}(z)^T \hat{z}) \]

\[ g(x) = g(z)(\Phi(\hat{x})) = \text{sign}(\overrightarrow{w}(z)^T \Phi(\hat{x})) \]
Must Choose $\Phi$ BEFORE Looking at the Data

- Rely on domain knowledge (feature engineering)
  - Handwriting digit recognition example

- Use common sets of feature transformation
  - Polynomial transformation
  - E.g., 2nd order Polynomial transformation
    - $\tilde{x} = (1, x_1, x_2)$, $\Phi_2(\tilde{x}) = (1, x_1, x_2, x_1 x_2, x_1^2, x_2^2)$
    - Plus: more powerful (contains circle, ellipse, hyperbola, etc)
    - Minus:
      - More computation/storage
      - Worse generalization error

The VC dimension of d-dim perceptron is $d+1$
Q-th Order Polynomial Transform

- Q-th Order Polynomial Transform
  - $\Phi_1(\vec{x}) = \vec{x}$
  - $\Phi_2(\vec{x}) = (\Phi_1(\vec{x}), x_1^2, x_1 x_2, x_1 x_3, ..., x_d^2)$
  - ...
  - $\Phi_Q(\vec{x}) = (\Phi_{Q-1}(\vec{x}), x_1^Q, x_1^{Q-1} x_2, ..., x_d^Q)$

- Each element in $\Phi_Q(\vec{x})$ is in the form of $\prod_{i=1}^{d} x_i^{a_i}$
  - where $\sum_{i=1}^{d} a_i \leq Q$, and $a_i$ is a non-negative integer

- Number of elements in $\Phi_Q(\vec{x})$: $\binom{Q+d}{Q}$ (including the initial 1)
Structural Hypothesis Sets

• Let $H_Q$ be the linear model for the $\Phi_Q(\hat{x})$ space

• Let $g_Q = \text{argmin}_{h \in H_Q} E_{in}(h)$
  • $H_1 \subseteq H_2 \subseteq H_3 \subseteq \ldots$
  • $d_{vc}(H_1) \leq d_{vc}(H_2) \leq \ldots$
  • $E_{in}(g_1) \geq E_{in}(g_2) \geq \ldots$
Today’s Lecture

The notes are not intended to be comprehensive. They should be accompanied by lectures and/or textbook. Let me know if you spot errors.
Overfitting

[Adapted from the slides by Malik Magdon-Ismail]
Setup of the Discussion

• Regression with polynomial transform
  • Input: 1-dimensional $x$
  • $\Phi_Q(x) = (1, x, x^2, x^3, \ldots, x^Q)$
  • $H_Q = \{h(x) = w_0 + w_1x + w_2x^2 + \cdots + w_Qx^Q\}$

• $Q$th-order polynomial fit
  • Solve linear regression on the $\Phi_Q(x)$ space using $H_Q$
  • Looking to minimize $E_{in}: g_Q = \text{argmin}_{h \in H_Q} E_{in}(h)$
A Simple Example

• Target $f$: 4$^\text{th}$ order function
• # data points: $N = 5$
• Small noise:
  • $y = f(x) + \epsilon$ with small $\epsilon$
• 4$^\text{th}$ order polynomial fit
  • $h(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$
  • Find $g_4 = \arg\min_h E_{in}(h)$
A Simple Example

• Target $f$: 4th order function
• # data points: $N = 5$
• Small noise:
  • $y = f(x) + \epsilon$ with small $\epsilon$
• 4th order polynomial fit
  • $h(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$
  • Find $g_4 = \arg\min_h E_{in}(h)$

Classical overfitting: $E_{in} = 0$, but lead to a large $E_{out}$

Fitting the noise instead of the target
What is Overfitting?

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

Overfitting
Going for lower and lower $E_{in}$ results in higher and higher $E_{out}$
Case Study: 2\textsuperscript{nd} vs 10\textsuperscript{th} Order Polynomial Fit
Which model do you choose for the left problem and why?
Target Function: $10^{\text{th}}$ Order $f$ with Noise

- Irony of two learners Red and Green
- Both know the target is $10^{\text{th}}$ order
- Red chooses $H_{10}$
- Green chooses $H_2$
- Green outperforms Red

<table>
<thead>
<tr>
<th></th>
<th>2nd Order</th>
<th>10th Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{\text{in}}$</td>
<td>0.050</td>
<td>0.034</td>
</tr>
<tr>
<td>$E_{\text{out}}$</td>
<td>0.127</td>
<td>9.00</td>
</tr>
</tbody>
</table>
Why is $H_2$ Better than $H_{10}$?

When $N$ is small, $E_{out}(g_{10}) \geq E_{out}(g_2)$
Which model do you choose for the right problem and why?

\[ H_2: \text{2}^{\text{nd}} \text{ order polynomial fit} \]
\[ H_{10}: \text{10}^{\text{th}} \text{ order polynomial fit} \]
Simpler $H$ is better even for complex target with no noise

<table>
<thead>
<tr>
<th>complex noiseless target</th>
<th>2nd Order</th>
<th>10th Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{in}$</td>
<td>0.029</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>$E_{out}$</td>
<td>0.120</td>
<td>7680</td>
</tr>
</tbody>
</table>
Is There Really “No Noise”? 

Simple $f$ with noise. 

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

Simple $f$ with noise. 

Complex $f$ with no noise.
Why is $H_2$ Better than $H_{10}$?

When $N$ is small, $E_{out}(g_{10}) \geq E_{out}(g_2)$
A Detailed Experiment

Study the level of noise and target complexity, and # data points \( N \)

\[
y = f(x) + \epsilon(x) = \sum_{q=0}^{Q_f} \alpha_q x^q + \epsilon(x)
\]

Noise level: variance \( \sigma^2 \) of \( \epsilon(x) \)
Target complexity: \( Q_f \)
Data set size: \( N \)
The Overfit Measure

• Fit the data set using $H_2$ and $H_{10}$
  • Let $g_2$ and $g_{10}$ be the learned hypothesis

• Overfit measure
  • $E_{out}(g_{10}) - E_{out}(g_2)$
  • This value is large is overfit happens
Overfit Measure: $E_{out}(g_{10}) - E_{out}(g_2)$

**Stochastic noise**

**Deterministic noise**

<table>
<thead>
<tr>
<th>Number of data points ↑</th>
<th>Noise ↑</th>
<th>Target complexity ↑</th>
<th>Overfitting ↓</th>
<th>Overfitting ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Noise:
The part of $y$ we **cannot** model
Stochastic Noise

We would like to learn from $\bigcirc$:

$$y_n = f(x_n)$$

Unfortunately, we only observe $\bigcirc$:

$$y_n = f(x_n) + \text{‘stochastic noise’}$$

\[\text{no one can model this}\]

\textbf{Stochastic Noise: fluctuations/measurement errors we cannot model.}
Stochastic Noise

We would like to learn from $y_n = f(x_n)$:

Unfortunately, we only observe $y_n = f(x_n) + \text{`stochastic noise'}$

\[\text{no one can model this}\]

**Stochastic Noise**: fluctuations/measurement errors we cannot model.
Deterministic Noise

We would like to learn from $y_n = h^*(x_n)$:

Unfortunately, we only observe $y_n = f(x_n) = h^*(x_n) + \text{‘deterministic noise’}$

$\mathcal{H}$ cannot model this

**Deterministic Noise:** the part of $f$ we cannot model.
Deterministic Noise

We would like to learn from $\mathcal{O}$:

$$y_n = h^*(x_n)$$

Unfortunately, we only observe $\mathcal{D}$:

$$y_n = f(x_n) = h^*(x_n) + \text{‘deterministic noise’}$$

$\mathcal{H}$ cannot model this

**Deterministic Noise:** the part of $f$ we cannot model.
Both sources of noises hurt learning

**Stochastic Noise**

- $y = f(x) + \text{stoch. noise}$

**Deterministic Noise**

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

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

**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 Bias-Variance Decomposition

\[ y = f(\hat{x}) + \epsilon \]

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

Stochastic Noise  Deterministic noise
How to Fight Overfitting

• VC Bound

\[ E_{out}(g) \leq E_{in}(g) + O\left(\sqrt{d_{vc} \frac{\ln N}{N}}\right) \]

• Fighting overfitting
  • Regularization
  • Validation
  • (The focus of the next two lectures)
VC Dimension of d-dim Perceptron
Recall the Definitions

• Shatter
  • $H$ shatters $(\vec{x}_1, ..., \vec{x}_N)$ if $|H(\vec{x}_1, ..., \vec{x}_N)| = 2^N$
  • $H$ can induce all label combinations for $(\vec{x}_1, ..., \vec{x}_N)$

• Break point
  • $k$ is a break point for $H$ if no data set of size $k$ can be shattered by $H$
  • $k$ is a break point for $H \leftrightarrow m_H(k) < 2^k$

• VC Dimension: $d_{vc}(H)$ or $d_{vc}$
  • The VC dimension of $H$ is the largest $N$ such that $m_H(N) = 2^N$
  • Equivalently, if $k^*$ is the smallest break point for $H$, $d_{vc}(H) = k^* - 1$
VC Dimension of d-dimension Perceptron

• Claim:
  • The VC Dimension of d-dim perceptron is $d + 1$

• How to prove it?
  1. Show that the VC dimension of d-dim perceptron $\geq d + 1$
  2. Show that the VC dimension of d-dim perceptron $\leq d + 1$
• To prove $d_{vc}(H) \geq d + 1$, what do we need to prove?

  A. There is a set of $d + 1$ points that can be shattered by $H$
  B. There is a set of $d + 1$ points that cannot be shattered by $H$
  C. Every set of $d + 1$ points can be shattered by $H$
  D. Every set of $d + 1$ points cannot be shattered by $H$
• To prove $d_{vc}(H) \geq d + 1$, what do we need to prove?

A. There is a set of $d + 1$ points that can be shattered by $H$
B. There is a set of $d + 1$ points that cannot be shattered by $H$
C. Every set of $d + 1$ points can be shattered by $H$
D. Every set of $d + 1$ points cannot be shattered by $H$
• To prove $d_{vc}(H) \geq d + 1$, what do we need to prove?
  A. There is a set of $d + 1$ points that can be shattered by $H$
  B. There is a set of $d + 1$ points that cannot be shattered by $H$
  C. Every set of $d + 1$ points can be shattered by $H$
  D. Every set of $d + 1$ points cannot be shattered by $H$

• To prove $d_{vc}(H) \leq d + 1$, what do we need to prove?
  A. There is a set of $d + 1$ points that can be shattered by $H$
  B. There is a set of $d + 2$ points that cannot be shattered by $H$
  C. Every set of $d + 2$ points can be shattered by $H$
  D. Every set of $d + 1$ points cannot be shattered by $H$
  E. Every set of $d + 2$ points cannot be shattered by $H$
• To prove $d_{vc}(H) \geq d + 1$, what do we need to prove?

  A. There is a set of $d + 1$ points that can be shattered by $H$
  B. There is a set of $d + 1$ points that cannot be shattered by $H$
  C. Every set of $d + 1$ points can be shattered by $H$
  D. Every set of $d + 1$ points cannot be shattered by $H$

• To prove $d_{vc}(H) \leq d + 1$, what do we need to prove?

  A. There is a set of $d + 1$ points that can be shattered by $H$
  B. There is a set of $d + 2$ points that cannot be shattered by $H$
  C. Every set of $d + 2$ points can be shattered by $H$
  D. Every set of $d + 1$ points cannot be shattered by $H$
  E. Every set of $d + 2$ points cannot be shattered by $H$
• To prove $d_{vc}(H) \geq d + 1$, what do we need to prove?
  There is a set of $d + 1$ points that can be shattered by $H$

• To prove $d_{vc}(H) \leq d + 1$, what do we need to prove?
  Every set of $d + 2$ points cannot be shattered by $H$