CSE 417T
Introduction to Machine Learning

Lecture 9
Instructor: Chien-Ju (CJ) Ho
Logistics

• Homework 2 is due on Feb 24 (Thursday)

• Return of Homework
  • We plan to return each homework within 2 weeks after the deadline
  • Regrade requests
    • You will have up to 7 days to submit regrade requests
      • the regrade period might be shortened if there are schedule constraints
    • We might check the entire homework for each request, so the grades might go down as well if we find new mistakes

• Exam 1: Mar 10 (Thursday)
  • Content: LFD Chap 1 to 5 (The entire hardcopy of the textbook)
  • Covid-permitting
    • Timed exam (75 min) during lecture time in the classroom
    • Closed-book exam with 2 letter-size cheat sheets allowed (4 pages in total)
      • No format limitations (it can be typed, written, or a combination)
Recap
Linear Models

• $H$ contains hypothesis $h(\hat{x})$ as **some function of** $\hat{w}^T \hat{x}$

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

This is why it’s called linear models
Logistic Regression

• Predict a probability
  • Interpreting $h(\hat{x}) \in [0,1]$ as the prob for $y = +1$ given $\hat{x}$

• Hypothesis set $H = \{h(\hat{x}) = \theta(\vec{w}^T \hat{x})\}$
  • $\theta(s) = \frac{e^s}{1 + e^s} = \frac{1}{1 + e^{-s}}$

• Algorithm
  • Find $g = \text{argmin}_{h \in H} E_{in}(h)$

• Two key questions
  • How to define $E_{in}(h)$?
  • How to perform the optimization (minimizing $E_{in}$)?
Define $E_{in}(\vec{w})$: Cross-Entropy Error

$$E_{in}(\vec{w}) = \frac{1}{N} \sum_{n=1}^{N} \ln(1 + e^{-y_n \vec{w}^T \vec{x}_n})$$

• Minimizing cross entropy error is the same as maximizing likelihood

• Likelihood: $Pr(D|\vec{w})$
  
  • $\vec{w}^* = \arg\max_{\vec{w}} Pr(D|\vec{w})$ (maximizing likelihood)
  
  • $\vec{w}^* = \arg\min_{\vec{w}} E_{in}(\vec{w})$ (minimizing cross-entropy error)
Optimizing $E_{in}(\vec{w})$: Gradient Descent

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

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

Works for functions where gradient exists everywhere
Nonlinear Transformation

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

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

\[ g(\tilde{x}) = g^{(z)}(\Phi(\tilde{x})) = \text{sign}(\tilde{w}^{(z)T} \Phi(\tilde{x})) \]
How to Choose Feature Transform \( \Phi \)

Something Seems Wrong!
Must choose \( \Phi \) BEFORE looking at the data.

Otherwise, you are doing “data snooping”

The hypothesis set \( H \) is as large as anything your brain can think of.
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.
Choose $\Phi$ Before Seeing Data

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

• Use common sets of feature transformation
  • Polynomial transformation
  • 2nd order Polynomial transformation
    • $\tilde{x} = (1, x_1, x_2)$
    • $\Phi_2(\tilde{x}) = (1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$
  • Pros: more powerful (contains circle, ellipse, hyperbola, etc)
  • Cons: 2-d => 5-d
    • More computation/storage
    • Worse generalization error

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

\[ \mathbf{x} = (1, x_1, \ldots, x_d) \]

- From 1-st order to Q-th order polynomial transform:
  - \( \Phi_1(\mathbf{x}) = \mathbf{x} \)
  - \( \Phi_2(\mathbf{x}) = (\Phi_1(\mathbf{x}), x_1^2, x_1x_2, x_1x_3, \ldots, x_d^2) \)
  - ...
  - \( \Phi_Q(\mathbf{x}) = (\Phi_{Q-1}(\mathbf{x}), x_1^Q, x_1^{Q-1}x_2, \ldots, x_d^Q) \)

- Number of elements in \( \Phi_Q(\mathbf{x}) \)
Q-th Order Polynomial Transform

- $\tilde{x} = (1, x_1, \ldots, x_d)$
- From 1-st order to Q-th order polynomial transform:
  - $\Phi_1(\tilde{x}) = \tilde{x}$
  - $\Phi_2(\tilde{x}) = (\Phi_1(\tilde{x}), x_1^2, x_1x_2, x_1x_3, \ldots, x_d^2)$
  - ...
  - $\Phi_Q(\tilde{x}) = (\Phi_{Q-1}(\tilde{x}), x_1^Q, x_1^{Q-1}x_2, \ldots, x_d^Q)$

- Number of elements in $\Phi_Q(\tilde{x})$
  - $\binom{Q + d}{Q}$
Structural Hypothesis Sets

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

![Diagram showing nested hypothesis sets $\mathcal{H}_0$, $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$, ...]

• Let $g_Q = \text{argmin}_{h \in H_Q} E_{in}(h)$
  - $H_0, H_1, H_2, ...$
  - $d_{vc}(H_0), d_{vc}(H_1), d_{vc}(H_2), ...$
  - $E_{in}(g_0), E_{in}(g_1), E_{in}(g_2), ...$
Structural Hypothesis Sets

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

• Let $g_Q = \text{argmin}_{h \in H_Q} E_{in}(h)$
  • $H_0 \subset H_1 \subset H_2 \ldots$
  • $d_{vc}(H_0) \leq d_{vc}(H_1) \leq d_{vc}(H_2) \ldots$
  • $E_{in}(g_0) \geq E_{in}(g_1) \geq E_{in}(g_2) \ldots$
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 $f$: 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 $g_4 = \text{argmin}_h E_{in}(h)$
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:
$2^{nd}$ vs $10^{th}$ Order Polynomial Fit
Which model would you choose for the left problem and why?

\[ H_2 : 2^{\text{nd}} \text{ order polynomial fit} \]
\[ H_{10} : 10^{\text{th}} \text{ order polynomial fit} \]
Target Function: 10\textsuperscript{th} Order $f$ with Noise

- Irony of two learners Red and Green
- Both \textbf{know} the target is 10\textsuperscript{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>$E_{in}$</td>
<td>0.050</td>
<td>0.034</td>
</tr>
<tr>
<td>$E_{out}$</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?

\( H_2: \) 2\textsuperscript{nd} order polynomial fit

\( H_{10}: \) 10\textsuperscript{th} order polynomial fit
Simpler $H$ is better even for complex target with **no noise**

<table>
<thead>
<tr>
<th></th>
<th>2nd Order</th>
<th>10th Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{\text{in}}$</td>
<td>0.029</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>$E_{\text{out}}$</td>
<td>0.120</td>
<td><strong>7680</strong></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 overfitting happens
Overfit Measure: $E_{out}(g_{10}) - E_{out}(g_{2})$
Noise:
The part of $y$ we cannot model
Stochastic Noise

We would like to learn from $\circ$:

$$y_n = f(x_n)$$

Unfortunately, we only observe $\bigcirc$:

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

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

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

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

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

We would like to learn from $\boxed{\circ}$:

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

Unfortunately, we only observe $\circ$:

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

$\mathcal{H}$ cannot model this

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

We would like to learn from $\bigcirc$:

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

Unfortunately, we only observe $\bigcirc$:

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

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

• **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 \( H \) if \( m_H(k) < 2^k \)

• **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 $H$
• 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 $H$

**Proof Sketch:**

1. Let’s construct a dataset of $d + 1$ points: $X = \begin{bmatrix} x^1_1 \\ \vdots \\ x^1_{d+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ 1 & 0 & 0 & \ldots & 0 & 1 \\ 1 & 0 & 0 & \ldots & 1 & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ 1 & 1 & 0 & \ldots & 0 & 0 \end{bmatrix}$; It’s easy to check that $X^{-1}$ exist

2. For any possible dichotomy $\vec{y}$, there exists a $\vec{w}$ such that $X\vec{w} = \vec{y}$, i.e., $\vec{w} = X^{-1}\vec{y}$

3. Therefore, $d$-dim perceptron can shatter $X$

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

Every set of $d + 2$ points cannot be shattered by $H$

**Proof Sketch:**

1. For every set of $d + 2$ points (in $d+1$ dimensions), there exists a point that can be written as linear combinations of the others.

2. Denote the point $\vec{x}_{d+2}$, we have $\vec{x}_{d+2} = \sum_{i=1}^{d+1} a_i \vec{x}_i$

3. Consider the dichotomy $(y_1, \ldots, y_{d+2}) = (\text{sign}(a_1), \ldots, \text{sign}(a_{d+1}), -1)$, we can show that no linear separator can generate this dichotomy (think about why).

4. Therefore, for every set of $d + 2$ points, there exist at least one dichotomy that $H$ cannot induce.