CSE 417T
Introduction to Machine Learning

Lecture 11
Instructor: Chien-Ju (CJ) Ho
Logistics

• Homework 2: Due **Feb 24** (Thu)
• Homework 3: Due **Mar 5** (Sat)
  • Keep track of your own late-day usages

• **Exam 1: Mar 10 (Thursday)**
  • Topics: LFD Chapters 1 to 5
  • 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)
  • Mar 8 (Tuesday) will be a review lecture
Recap
Overfitting and Its Cures

- **Overfitting**
  - Fitting the data more than is warranted
  - Fitting the noise instead of the pattern of the data
  - Decreasing $E_{\text{in}}$ but getting larger $E_{\text{out}}$
  - When $H$ is too strong, but $N$ is not large enough

- **Regularization**
  - Intuition: Constrain $H$ to make overfitting less likely to happen

- **Validation**
  - Intuition: Reserve data to estimate $E_{\text{out}}$
Regularization (Constrain $H$)

- Weight decay

\[ H(C) = \{ h \in H_Q \text{ and } \vec{w}^T \vec{w} \leq C \} \]

- Algorithm: Find $g \in H(C)$ such that $g \approx f$

Constrained optimization

\[
\begin{align*}
\text{minimize} & \quad E_{in}(\vec{w}) \\
\text{subject to} & \quad \vec{w}^T \vec{w} \leq C
\end{align*}
\]

Unconstrained optimization

\[
\begin{align*}
\text{minimize} & \quad E_{in}(\vec{w}) + \frac{\lambda_c}{N} \vec{w}^T \vec{w} \\
\text{Augmented error}
\end{align*}
\]
Augmented Error

\[ E_{aug}(h, \lambda, \Omega) = E_{in}(\overline{w}) + \frac{\lambda}{N} \Omega(h) \]

• Key components
  • \( \Omega \): Regularizer
  • \( \lambda \): Amount of regularization

• Does the form look familiar? Recall in the VC Theory (treating \( \delta \) as a constant)
  • \( E_{out}(g) \leq E_{in}(g) + O \left( \sqrt{d_{vc} \frac{\ln N}{N}} \right) \)

• What the impacts of picking \( \Omega \) and \( \lambda \)?
Summary of Regularization

• Regularization is everywhere in machine learning

• Two main ways of thinking about regularization
  • Constrain $H$ to make overfitting less likely to happen
    • Will discuss more regularization methods in the 2nd half of the semester
    • Pruning for decision trees, early stopping / dropout for neural networks, etc
  
  • Define augmented error $E_{aug}$ to better approximate $E_{out}$
    • $E_{aug}(h, \lambda, \Omega) = E_{in}(h) + \frac{\lambda}{N} \Omega(h)$

• We show the equivalence of the two for weight decay
  • The conceptual equivalence is general with Lagrangian relaxation (will cover later in the semester)
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.
Prevent Overfitting

\[ E_{out}(g) = E_{in}(g) + \text{overfit penalty} \]

- Regularization
  - Choose a regularizer \( \Omega \) to approximate the penalty

- Validation
  - Directly estimate \( E_{out} \) (The goal of learning is to minimize \( E_{out} \))
Review of Test Set (Estimate $E_{out}$)

• Out-of-sample error $E_{out}(g) = \mathbb{E}_{\tilde{x}}[e(g(\tilde{x}), y)]$
  • Key: $\tilde{x}$ need to be out of sample (i.e., not in training)

• Test set $D_{test} = \{(\tilde{x}_1, y_1), ..., (\tilde{x}_K, y_K)\}$
  • Reserve $K$ data points
  • None of the data points in test set can be involved in training

• Using the data in test set to estimate $E_{out}$
  • Since all data points in $D_{test}$ are out of sample
In HW2, you are asked to perform “normalization” on the training/test datasets. How should you do it?

1. Calculate the mean/variance of the combined data. Normalize them using the overall mean/variance.

2. Calculate the means/variances of the training and test datasets separately. Normalize them using their respective mean/variance.

3. Calculate the mean/variance of the training dataset. Normalize both datasets using the training mean/variance.
Short Discussion on HW2

• In HW2, you are asked to perform “normalization” on the training/test datasets. How should you do it?

1. Calculate the mean/variance of the combined data. Normalize them using the overall mean/variance.

2. Calculate the means/variances of the training and test datasets separately. Normalize them using their respective mean/variance.

3. Calculate the mean/variance of the training dataset. Normalize both datasets using the training mean/variance.

Two important properties we want to preserve
1. Training and test data are drawn from the same distribution.
2. Test data is never used in training.
Test Set

• Test set \( D_{test} = \{(\hat{x}_1, y_1), \ldots, (\hat{x}_K, y_K)\} \)

• For a \( g \) learned using only the training dataset
  • \( g \) is a “fixed” hypothesis for \( D_{test} \)

• Let \( E_{test}(g) = \frac{1}{K} \sum_{k=1}^{K} e(g(\hat{x}_k), y_k) \)
  • \( E_{test}(g) \) is an unbiased estimate of \( E_{out}(g) \)
    • \( \mathbb{E}[E_{test}(g)] = \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}[e(g(\hat{x}_k), y_k)] = E_{out}(g) \)
  • Single-hypothesis Hoeffding bound applies
    • \( E_{out}(g) \leq E_{test}(g) + O\left(\frac{1}{\sqrt{K}}\right) \)
Where are Test Set From?

• Given a data set $D$ of $N$ points
  • $D = D_{\text{train}} \cup D_{\text{test}}$
  • Reserving $K$ points for test set means we only have $N - K$ points for training

• Effect of the choice of $K$
Where are Test Set From?

- Given a data set $D$ of $N$ points
  - $D = D_{\text{train}} \cup D_{\text{test}}$
  - Reserving $K$ points for test set means we only have $N - K$ points for training

- Effect of the choice of $K$

Rule of Thumb: $K^* = \frac{N}{5}$
Utilizing the Whole $\mathcal{D}$

- Process:
  - $\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}}$ where $|\mathcal{D}_{\text{test}}| = K, |\mathcal{D}_{\text{train}}| = N - K$
  - Learn some hypothesis $g^-$ using only $\mathcal{D}_{\text{train}}$
  - Estimate $E_{\text{out}}(g^-)$ using $\mathcal{D}_{\text{test}}$

- Can we do better than $g^-$?
  - Yes! Learn $g$ using the entire $\mathcal{D}$; return $g$ and $E_{\text{test}}(g^-)$

- Generally (Informal, not theoretically proven)
  - Training on more data leads to better learned hypothesis
  - $E_{\text{out}}(g) \leq E_{\text{out}}(g^-)$
Validation: Beyond Test Set

What if we want to estimate $E_{out}$ multiple times?
Validation: Beyond Test Set

• Model selection:
  • Should I use linear models or decision trees?
  • Should I set the regularization parameter $\lambda$ to 0.1, 0.01, or 0.001?
    • A model with different $\lambda$ can be considered as different model

• Validation set
  • $D = D_{train} \cup D_{val}$

• Key difference to the test set
  • $D_{val}$ could be used multiple times for model selection
  • We need to account for the multiple usages of $D_{val}$
Model Selection

• Which model should we choose?
Model Selection using Validation

• Which model should we choose?

Choose $H_m^*$ such that $E_{\text{val}}(g_{m^*}) \leq E_{\text{val}}(g_m)$ for all $m$
Question...

• Which of the following is true?

(a) $\mathbb{E}[\text{Eval}(g_{m^*})] = E_{\text{out}}(g_{m^*})$

(b) $\mathbb{E}[\text{Eval}(g_{m^*})] \leq E_{\text{out}}(g_{m^*})$

(c) $\mathbb{E}[\text{Eval}(g_{m^*})] \geq E_{\text{out}}(g_{m^*})$

Choose $H_{m^*}$ such that $E_{\text{val}}(g_{m^*}) \leq E_{\text{val}}(g_m)$ for all $m$. 
Question...

• Which of the following is true?

(a) \( \mathbb{E}[E_{\text{val}}(g_{m}^{*-})] = E_{\text{out}}(g_{m}^{*-}) \)

(b) \( \mathbb{E}[E_{\text{val}}(g_{m}^{*-})] \leq E_{\text{out}}(g_{m}^{*-}) \)

(c) \( \mathbb{E}[E_{\text{val}}(g_{m}^{*-})] \geq E_{\text{out}}(g_{m}^{*-}) \)

Choose \( H_{m}^{*} \) such that \( E_{\text{val}}(g_{m}^{*-}) \leq E_{\text{val}}(g_{m}^{*-}) \) for all \( m \)

Equivalent to use \( D_{\text{val}} \) to choose from \( H = \{ g_{1}^{*-}, ..., g_{M}^{*-} \} \)

\[
E_{\text{out}}(g_{m}^{*-}) \leq E_{\text{val}}(g_{m}^{*-}) + O\left(\sqrt{\frac{\ln M}{K}}\right)
\] => Hoeffding Bound for Multiple Hypothesis
Question...

• Which of the following is true?

(a) $\mathbb{E}[E_{val}(g_{m}^*)] = E_{out}(g_{m}^*)$

(b) $\mathbb{E}[E_{val}(g_{m}^*)] \leq E_{out}(g_{m}^*)$

(c) $\mathbb{E}[E_{val}(g_{m}^*)] \geq E_{out}(g_{m}^*)$

Equivalent to use $D_{val}$ to choose from $H = \{g_1^-, ..., g_M^\}$

$$E_{out}(g_{m}^*) \leq E_{val}(g_{m}^*) + O\left(\sqrt{\frac{\ln M}{K}}\right)$$ => Hoeffding Bound for Multiple Hypothesis
Utilizing the Whole $D$

$g_m^*$: the hypothesis minimizes in-sample error over $\{H_1, \ldots, H_M\}$
When a validation set is not used for model selection (i.e., used only once), it is essentially a test set.

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Relationship to $E_{out}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{in}$</td>
<td></td>
</tr>
<tr>
<td>$E_{val}$</td>
<td></td>
</tr>
<tr>
<td>(when used for model selection)</td>
<td></td>
</tr>
<tr>
<td>$E_{test}$</td>
<td></td>
</tr>
<tr>
<td>Outlook</td>
<td>Relationship to $E_{out}$</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>$E_{in}$</td>
<td>Incredibly optimistic</td>
</tr>
<tr>
<td>$E_{val}$ (when used for model selection)</td>
<td>Slightly optimistic</td>
</tr>
<tr>
<td>$E_{test}$</td>
<td>Unbiased</td>
</tr>
<tr>
<td>Outlook</td>
<td>Relationship to $E_{out}$</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>$E_{in}$</td>
<td>Incredibly optimistic</td>
</tr>
<tr>
<td>$E_{val}$ (when used for model selection)</td>
<td>Slightly optimistic</td>
</tr>
<tr>
<td>$E_{test}$</td>
<td>Unbiased</td>
</tr>
</tbody>
</table>

Note that the outlook comparisons are “in expectation”.
If you only get one “draw” of $D_{train}, D_{val}, D_{test}$, you cannot say anything “for certain”.

Remember that ML results are under the condition “with high probability”.
The Dilemma When Choosing $K$

• The main ideas behind validation

\[ E_{\text{out}}(g) \approx E_{\text{out}}(g^-) \approx E_{\text{val}}(g^-) \]
The Dilemma When Choosing $K$

- The main ideas behind validation

Want small $K$  
(didn’t sacrifice too much training data)

Want large $K$  
($E_{val}$ estimates $E_{out}$ well)

$$E_{out}(g) \approx E_{out}(g^-) \approx E_{val}(g^-)$$
Leave-One-Out Cross Validation (LOOCV)

Getting the best of both worlds

Intuition: Setting $K = 1$ but do it many times...
Illustrative Example

\[ E_{cv} = \frac{1}{3} (d_1^2 + d_2^2 + d_3^2) \]
Properties of LOOCV

• LOOCV is unbiased (If *not* used for model selection)
  • \( E_{CV} \) is an unbiased estimator of \( \bar{E}_{out}(N - 1) \)
    (expected \( E_{out} \) when learning on \( N - 1 \) points)

• The “effective number” of examples in \( E_{CV} \) estimation is high for LOOCV

• However, LOOCV is computationally expensive
  • Need to train \( N \) models, each on \( N - 1 \) points
V-Fold Cross Validation

• Split $D$ into $V$ equally sized data sets: $D_1, D_2, \ldots, D_V$
  
• Let $g_i^-$ be the hypothesis learned using all data sets except $D_i$

• Let $e_i = E_{val}(g_i^-)$ where the validation uses data set $D_i$

• The $V$-fold cross validation error is $\frac{1}{V} \sum_{i=1}^{V} e_i$

• Practical rule of thumb: $V = 10$
Three Learning Principles
Occam’s Razor

Sampling Bias

Data Snooping
Occam’s Razor

“An explanation of the data should be made as simple as possible, but no simpler.”  -- Einstein?

“entia non sunt multiplicanda praeter necessitatem”
(entities must not be multiplied beyond necessity)
-- William of Occam

“trimming down”
unnecessary explanation
The simplest model that fits the data is also the most plausible.

What does it mean to be simple?

Why is simple better?
Simple Model?

• For a hypothesis set $H$ to be simple
  • # dichotomies it can generate is small
  • VC Dimension is small

• For a hypothesis $h$ to be simple
  • lower order polynomial
  • smaller weights (think about the regularization)
  • easy to describe?
  • fewer number of parameters (fewer bits to describe)
Simple Model?

Connection:

A hypothesis set with *simple* hypotheses should be *simple*

Consider a hypothesis $h$ can be specified by $\ell$ bits

$\Rightarrow H$ contains all such $h$

$\Rightarrow$ The size of $H$ is $2^{\ell}$

Simple: small model complexity / VC dimension / size of hypothesis set
Why is Simple Better?

simple -> small VC dimension -> good generalization, less overfitting, ...

Simple $\mathcal{H}$

$\Rightarrow$ small growth function $m_\mathcal{H}(N)$

$\Rightarrow$ if data labels are generated randomly, the probability of fitting perfectly is?

$$\frac{m_\mathcal{H}(N)}{2^N}$$

$\Rightarrow$ more significant when fit really happens
Occam’s Razor
Sampling Bias
Data Snooping
1948 US Presidential Election

- Truman vs. Dewey
- Chicago Daily Tribune decided to run a phone poll of how people voted
Truman
What happened?

One explanation: we cannot claim anything for certain.

However, there are bigger issues here...

- Phones are expensive in 1948...
- Dewey was more favored in rich populations

- Imagine you are polling from people in DC/Texas/NY to predict who will win the presidential election...
Sampling Bias

If the data is sampled in a biased way, learning will produce a similarly biased outcome.