CSE 417T
Introduction to Machine Learning

Lecture 12
Instructor: Chien-Ju (CJ) Ho
Logistics: Reminders

• HW 2: Feb 24, 2020 (Monday)
  • Reserve time for submissions
  • No extensions will be given for last-minute technical reasons

• Exam 1: March 3, 2020 (Tuesday)
  • In-class exam (the same time/location as the lecture)
  • Exam duration: 75 minutes
  • Planned exam content: LFD Chapter 1 to 5
  • Check seat assignments on Piazza the night before the exam
  • More details in the Slides on Feb 18
Recap
Overfitting and Its Cures

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

• Regularization
  • Intuition: Constraining $H$ to make overfitting less likely to happen

• Validation
  • Intuition: Reserve data to estimate $E_{out}$
Regularization (Constraining $H$)

- 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

\[
\text{minimize } E_{in}(\vec{w}) \\
\text{subject to } \vec{w}^T\vec{w} \leq C
\]

Unconstrained optimization

\[
\text{minimize } E_{in}(\vec{w}) + \frac{\lambda_c}{N}\vec{w}^T\vec{w}
\]

Augmented error
Augmented Error

• Define augmented error
  
  \[ E_{aug}(\vec{w}) = E_{in}(\vec{w}) + \frac{\lambda_C}{N} \vec{w}^T \vec{w} \]
  
  • Algorithm: Find \( \vec{w}^* = \arg\min E_{aug}(\vec{w}) \)

• A bit more discussion
  
  • When \( C \to \infty, \lambda_C = 0 \)
  
  • Smaller \( C \) (stronger constraints)
    
    • \( \Rightarrow \) larger \( \lambda_C \)
    
    • \( \Rightarrow \) smaller \( H \)
    
    • \( \Rightarrow \) stronger regularization
  
  • Use \( \lambda_C \) to tune the level of regularization

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

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

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

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

• If we pick the right \( \Omega \), \( E_{aug} \) can be a good proxy for \( E_{out} \)
How to Pick the Right $\Omega$

- Intuition: pick $\Omega$ that leads to “smoother” hypothesis
  - Overfitting is due to noise
  - Informally, noise is generally ”high frequency”

- Computation: prefer $\Omega$ that makes the optimization easier (e.g., convex/differentiable)
  - Similar to picking the error measure

- We might have some other objective in mind
  - Ex: L-1 regularizer leads to weight vectors with more 0s
Brief Lecture Notes Today

The notes are not intended to be comprehensive. They should be accompanied by lectures and/or textbook. Let me know if you spot errors.
More Discussion on Regularization
Why $\vec{w}^T \vec{w}$ is Called Weight Decay

• Run gradient descent on $E_{aug}(\vec{w}) = E_{in}(\vec{w}) + \lambda_c \vec{w}^T \vec{w}$

• The update rule would be

\[
\vec{w}(t + 1) \leftarrow \vec{w}(t) - \eta \nabla_{\vec{w}} E_{aug}(\vec{w}(t))
\]

\[
\Rightarrow \vec{w}(t + 1) \leftarrow (1 - 2\eta \lambda_c)\vec{w}(t) - \eta \nabla_{\vec{w}} E_{in}(\vec{w}(t))
\]

We are **decaying** the weights first, then do the update
Linear Regression with Weight Decay

\[ E_{aug}(\vec{w}) = E_{in}(w) + \frac{\lambda_c}{N} \vec{w}^T \vec{w} = \frac{1}{N} \|X\vec{w} - \vec{y}\|^2 + \frac{\lambda_c}{N} \vec{w}^T \vec{w} \]

Solve \( \nabla_{\vec{w}} E_{aug}(\vec{w}) \big|_{\vec{w}=\vec{w}_{reg}} = 0 \), we get

1. \( \frac{2}{N} (X^TX\vec{w}_{reg} - X^T\vec{y} + \lambda_c \vec{w}_{reg}) = 0 \)
2. \( (X^TX + \lambda_c I)\vec{w}_{reg} = X^T\vec{y} \)
3. \( \vec{w}_{reg} = (X^TX + \lambda_c I)^{-1}X^T\vec{y} \)

Notation: \( I \) is an identity matrix: only the elements in the diagonals are 1, and all others are 0.

This is called “Ridge Regression” in statistics.
Effect of Regularization (Different $\lambda$)

- Minimizing $E_{in}(\vec{w}) + \frac{\lambda}{N} \vec{w}^T \vec{w}$ with different $\lambda$

\[ \lambda = 0 \] \hspace{1cm} \[ \lambda = 0.0001 \] \hspace{1cm} \[ \lambda = 0.01 \] \hspace{1cm} \[ \lambda = 1 \]

Overfitting → → Underfitting
Overfitting and Underfitting

Need to pick the right $\lambda$:
Using validation: Focus of this lecture
Variations on Weight Decay (Different $\Omega$)

Uniform Weight Decay

Low Order Fit

Weight Growth!

\[
\sum_{q=0}^{Q} w_q^2
\]

\[
\sum_{q=0}^{Q} qw_q^2
\]

\[
\sum_{q=0}^{Q} \frac{1}{w_q^2}
\]
How to Pick the Right $\Omega$

• As discussed earlier
  • Intuition: pick $\Omega$ that leads to “smoother” hypothesis
    • Overfitting is due to noise
    • Informally, noise is generally “high frequency”
  • Computation: prefer $\Omega$ that makes the optimization easier (e.g., convex/differentiable)
    • Similar to picking the error measure
  • We might have some other objective in mind
    • Ex: L-1 regularizer leads to weight vectors with more 0s

• What if we pick the wrong $\Omega$ (weight growth)
  • We might still fix it by picking the right $\lambda$ – using validation
Summarizing Regularization

- Regularization is everywhere in machine learning

- Two main ways of thinking about regularization
  - Constraining $H$ 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 $E_{out}$
    - $E_{aug}(h, \lambda, \Omega) = E_{in}(\vec{w}) + \frac{\lambda}{N} \Omega(h)$

- We show the equivalence of the two for weight decay
  - The conceptual equivalence is general with Lagrangian relaxation (will cover later in the semester)
Validation
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 real goal of learning is to minimize \( E_{out} \) )
Test Set (Want to 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

• Test set $D_{test} = \{(\tilde{x}_1, y_1), ..., (\tilde{x}_K, y_K)\}$
  • Reserve $K$ data points used to estimate $E_{out}$
  • 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
Test Set

• Test set $D_{test} = \{ (\vec{x}_1, y_1), ..., (\vec{x}_K, y_K) \}$
• For a $g$ learned using only training set

• Let $E_{test}(g) = \frac{1}{K} \sum_{k=1}^{K} e(g(\vec{x}_k), y_k)$

  • $E_{test}(g)$ is an unbiased estimate of $E_{out}(g)$
    • $\mathbb{E}[E_{test}(g)] = \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}[e(g(\vec{x}_k), y_k)] = E_{out}(g)$

  • Single hypothesis Hoeffding bound applies
    • $E_{out}(g) \leq E_{test}(g) + O\left(\sqrt{\frac{1}{K}}\right)$
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 $D$

• Process:
  • $D = D_{train} \cup D_{test}$ where $|D_{test}| = K$, $|D_{train}| = N - K$
  • Learn some hypothesis $g^-$ using only $D_{train}$
  • Estimate $E_{out}(g^-)$ using $D_{test}$
  • Let $g$ be the hypothesis that would be learned using $D$

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

• What if we want to estimate $E_{out}$ multiple times?

• 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: We need to account for the multiple usage of $D_{val}$
Model Selection

• Which model should we choose?
Model Selection using Validation

• Which model should we choose?

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

Key: $D_{val}$ is used $M$ times
Question…

• Which of the following is true?

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

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

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

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

• Which of the following is true?

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

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

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

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

Equivalent to use $D_{val}$ to choose from $H = \{g_{1}^{-}, ..., g_{M}^{-}\}$

$E_{out}(g_{m}^{\ast}) \leq E_{\text{val}}(g_{m}^{\ast}) + O\left(\sqrt{\frac{\ln M}{K}}\right)$ => 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 \( H = \{g_1^-, \ldots, g_M^-\} \)

\[ E_{out}(g_{m^*}^-) \leq E_{val}(g_{m^*}^-) + O\left(\sqrt{\frac{\ln M}{N}}\right) \quad \Rightarrow \quad \text{Hoeffding Bound for Multiple Hypothesis} \]
Utilizing the Whole $D$

$g_{\hat{m}}$: the hypothesis minimizes in-sample error over $\{H_1, \ldots, H_M\}$
<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>$E_{test}$</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>$E_{val}$</td>
<td>Slightly optimistic</td>
</tr>
<tr>
<td>$E_{test}$</td>
<td>Unbiased</td>
</tr>
</tbody>
</table>
### Outlook vs. Relationship to $E_{out}$

<table>
<thead>
<tr>
<th>$E_{in}$</th>
<th>Incredibly optimistic</th>
<th>VC-bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{val}$</td>
<td>Slightly optimistic</td>
<td>Hoeffding’s bound (multiple hypotheses)</td>
</tr>
<tr>
<td>$E_{test}$</td>
<td>Unbiased</td>
<td>Hoeffding’s bound (single hypothesis)</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

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

$$E_{out}(g) \approx E_{out}(g^-) \approx E_{val}(g^-)$$

Want small $K$
(didn’t sacrifice too much training data)
Leave-One-Out Cross Validation (LOOCV)

Getting the best of the both world

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: $D_1, D_2, ..., D_V$
  - 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: $V = 10$