CSE 417T
Introduction to Machine Learning

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

• Homework 2: due on Oct 7 (Friday)

• Exam 1: October 27 (Thursday)
  • Topics: LFD Chapters 1 to 5
  • Timed exam (75 min) during lecture time
  • Location TBD
  • 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)

• Homework 3 will be posted later today or tomorrow
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 \( H \)
  • Example: Weight decay \( H(C) = \{ h \in H_Q \text{ and } w^T w \leq C \} \)
  • Finding \( g \) => Constrained optimization

• Define augmented error
  • \( E_{aug}(h, \lambda, \Omega) = E_{in}(h) + \frac{\lambda}{N} \Omega(h) \)
  • Finding \( g \) => Unconstrained optimization

• The two interpretations are conceptually equivalent in a lot of cases.

• Understand the impacts of choosing \( \Omega \) and \( \lambda \)
Validation

• Reserve data to estimate $E_{out}$

Model Selection

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”

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Relationship to $E_{out}$</th>
</tr>
</thead>
<tbody>
<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>

• Cross Validation
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.
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 \textit{simple} hypotheses should be \textit{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
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.
What can we do...

Make sure the training and test distributions are as close as possible...
- Example: importance weighting

Not always possible....
- If you don’t have access to some region of points in training, but they appear in the testing distribution
Credit card example

• Determine whether to approve credit cards given applicants’ financial information

• Banks have lots of data:
  • Customer information
  • Whether they are good customers or not

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>32 years</td>
</tr>
<tr>
<td>gender</td>
<td>male</td>
</tr>
<tr>
<td>salary</td>
<td>40,000</td>
</tr>
<tr>
<td>debt</td>
<td>26,000</td>
</tr>
<tr>
<td>years in job</td>
<td>1 year</td>
</tr>
<tr>
<td>years at home</td>
<td>3 years</td>
</tr>
</tbody>
</table>

• Are there any issues here?

Approve for credit?
We will spend 1~2 lectures towards the end of the semester to talk about various ethical considerations of ML.
Occam’s Razor
Sampling Bias
Data Snooping
Data Snooping

If a data set has affected any step in the learning process, its ability to assess the outcome has been compromised.
Shouldn’t look at the data before selecting $H$
A Subtle Example

• Predict US Dollar vs. British Pound
  • $\hat{x}$: the change for the previous 20 days
  • $y$: the change in the 21th day

• Normalize data

• Split data $D = D_{\text{train}} \cup D_{\text{test}}$

• Where does snooping happen?
  • The normalization “looks at” $D_{\text{test}}$

• How should you perform normalization in Q1 of HW2?
Reuse of a Data Set

- Try one model after another on the same data set, you will eventually succeed.

“If you torture the data long enough, it will confess”

- VC dimension of the total learning models
- May even include what others tried (e.g., if you read their paper...)
- p-hacking...
JELLY BEANS CAUSE ACNE!
SCIENTISTS! INVESTIGATE!
BUT WE'RE PLAYING MINECRAFT!
...FINE.

WE FOUND NO LINK BETWEEN JELLY BEANS AND ACNE (\( p > 0.05 \)).

THAT SETTLES THAT.
I HEAR IT'S ONLY A CERTAIN COLOR THAT CAUSES IT.
SCIENTISTS!
BUT MINECRAFT!

From xkcd, by Randall Munroe: http://xkcd.com/882
What Should We Do...

Avoid data snooping
- Strict discipline
- E.g., be **honest** and lock the test data

Account for data snooping
- Measure how much data is contaminated
- E.g., what we discussed in validation
Content of Exam 1 Till Here
Course Plan

• Foundations
  • What’s machine learning
  • Feasibility of learning
  • Generalization
  • Linear models
  • Non-linear transformations
  • Overfitting and how to avoid it
    • Regularization
    • Validation

• Techniques
  • Decision tree
  • Ensemble learning
    • Bagging and random forest
    • Boosting and Adaboost
  • Nearest neighbors
  • Support vector machine
  • Neural networks
  • ...
Decision Tree
Decision Tree Hypothesis

Let $\mathbf{x} = (\text{annual income}, \text{have debt})$.

- $\mathbf{x} \geq 100k$ 
  - Approve
- $\mathbf{x} \geq 20k < 100k$ 
  - have debt?
    - yes 
      - Deny
    - no 
      - Approve
- $\mathbf{x} < 20k$ 
  - Deny

Let $y \in \{\text{approve, deny}\}$.
Decision Tree Hypothesis

Pros

- Easy to interpret (interpretability is getting attention and is important in many domains)
- Can handle multi-type data (Numerical, categorical...)
- Easy to implement (Bunch of if-else rules)

Cons

- Generally speaking, bad generalization
- VC dimension is infinity
- High variance (small change of data leads to very different hypothesis)
- Easily overfit

Why we care?

- One of the classical model
- Building block for other models (e.g., random forest)

Credit Card Approval Example

- Annual Income
  - \( \geq 100k \)
  - \( \geq 20k \)
  - \(< 100k \)
  - \(< 20k \)

- have debt?
  - yes
  - no

- Approve
  - Deny

- Deny
  - Approve
Decision Tree Hypothesis

- **Pros**
  - Easy to interpret (interpretability is getting attention and is important in many domains)
  - Can handle multi-type data (Numerical, categorical, ...)
  - Easy to implement (Bunch of if-else rules)

- **Cons**
  - Generally speaking, bad generalization
  - VC dimension is infinity
  - High variance (small change of data leads to very different hypothesis)
  - Easily overfit

- **Why we care?**
  - One of the classical models
  - Building block for other models (e.g., random forest)

Credit Card Approval Example

- Annual Income
  - ≥ 100k
  - ≥ 20k
  - < 100k
  - < 20k

- have debt?
  - yes
  - no

- Approve
- Deny

Deny
- Approve
- Deny

yes
- no

Approve
- Deny

Learning Decision Tree from Data

• Given dataset $D$, how to learn a decision tree hypothesis?

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>-1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
</tr>
<tr>
<td>-1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>-1</td>
<td>+1</td>
<td>-1</td>
<td>+1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>+1</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

• Potential approach
  • Find $g = \arg \min_{h \in H} E_{in}(h)$

• Multiple decision trees with zero $E_{in}$

Which one do you think might generalize better?
Learning Decision Tree from Data

• Conceptual intuition to deal with overfitting
  • Regularization: Constrain $H$

• Informally,

\[
\begin{align*}
\text{minimize} & \quad E_{in} \\
\text{subject to} & \quad \text{size}(\text{tree}) \leq C
\end{align*}
\]

• This optimization is generally computationally intractable.
• Most decision tree learning algorithms rely on heuristics to approximate the goal.
Template of Greedy-Based Decision Tree Algorithm

- DecisionTreeLearn($D$): Input a dataset $D$, output a decision tree hypothesis
  - Create a root node
  - If termination conditions are met
    - return a single node tree with leaf prediction based on $D$
  - Else: Greedily find a feature $A$ (assigned as root) to split according to split criteria
    - For each possible value $v_i$ of $A$
      - Let $D_i$ be the dataset containing data with value $v_i$ for feature $A$
      - Create a subtree DecisionTreeLearn($D_i$) that being the child of root

- Most decision tree learning algorithms follow this template, but with different choices of heuristics
Example

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>-1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
</tr>
<tr>
<td>-1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>-1</td>
<td>+1</td>
<td>-1</td>
<td>+1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>+1</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

DecisionTreeLearn($D$)
Create a root node
If termination conditions are met
    return a single node tree with leaf prediction based on $D$
Else: Greedily find a feature $A$ (assigned as root) to split according to split criteria
For each possible value $v_i$ of $A$
    Let $D_i$ be the dataset containing data with value $v_i$ for feature $A$
    Create a subtree DecisionTreeLearn($D_i$) that being the child of root

Termination conditions not met
Find a feature to split

Leaf prediction +1

Don’t terminate

Find next feature to split
Example Heuristics

DecisionTreeLearn(\(D\))
Create a root node
If termination conditions are met
    return a single node tree with leaf prediction based on \(D\)
Else: Greedily find a feature \(A\) to split according to split criteria
For each possible value \(v_i\) of \(A\)
    Let \(D_i\) be the dataset containing data with value \(v_i\) for feature \(A\)
    Create a subtree DecisionTreeLearn(\(D_i\)) that being the child of root

• Termination conditions
  • When the dataset is empty
  • When all labels are the same
  • when all features are the same
  • When the depth of the tree is too deep
  • ...

• Leaf predictions
  • Majority voting
  • Average (for regression)
  • ...

• Split criteria?
Split Criteria

• Which feature would you choose to split?

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>-1</td>
<td>+1</td>
</tr>
<tr>
<td>-1</td>
<td>+1</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
Split Criteria

• Which feature would you choose to split?

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>-1</td>
<td>+1</td>
</tr>
<tr>
<td>-1</td>
<td>+1</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

• Want the tree to be “smaller”
  • Intuition: choose the one that the labels in the subtrees are more “pure”
  • Example: choose the one maximizing information gain => ID3 Algorithm
Brief Intro to Information Entropy

• Assume there are $K$ possible labels
• Entropy:
  • $H(D) = \sum_{i=1}^{K} p_i \log_2 \frac{1}{p_i}$
  • $p_i$: ratio of points with label $i$ in the data

By definition
$0 \log_2 1 = 0; 1 \log_2 1 = 0$
Brief Intro to Information Entropy

• Assume there are $K$ possible labels

• Entropy:
  
  • $H(D) = \sum_{i=1}^{K} p_i \log_2 \frac{1}{p_i}$
  
  • $p_i$: ratio of points with label $i$ in the data

• Binary case with $K = 2$

By definition

$0 \log_2 \frac{1}{0} = 0 ; \quad 1 \log_2 \frac{1}{1} = 0 $
Brief Intro to Information Entropy

• Assume there are $K$ possible labels

• Entropy:
  
  \[ H(D) = \sum_{i=1}^{K} p_i \log_2 \frac{1}{p_i} \]

  • $p_i$: ratio of points with label $i$ in the data

• Binary case with $K = 2$

  • Interpretations of entropy
    • Expected # bit to encode a distribution

  • Higher entropy
    • data is less “pure”

  • ”pure” data $\Rightarrow$ all labels are +1 or -1 $\Rightarrow$ entropy = 0

  • Want to choose splits that lead to pure data, i.e., lower entropy

By definition

\[ 0 \log_2 \frac{1}{0} = 0; \ 1 \log_2 \frac{1}{1} = 0 \]
ID3: Using Information Gain as Selection Criteria

• Information gain of choosing feature $A$ to split
  
  \[ \text{Gain}(D, A) = H(D) - \sum_i \frac{|D_i|}{|D|} H(D_i) \]  
  [The amount of decrease in entropy]

• ID3: Choose the split that maximize $\text{Gain}(D, A)$

**DecisionTreeLearn**($D$)

Create a root node

If termination conditions are met
  
  return a single node tree with leaf prediction based on $D$

Else: Greedily find a feature $A$ to split according to split criteria

For each possible value $v_i$ of $A$

Let $D_i$ be the dataset containing data with value $v_i$ for feature $A$

Create a subtree DecisionTreeLearn($D_i$) that being the child of root

• ID3 termination conditions
  
  • If all labels are the same
  
  • If all features are the same
  
  • If dataset is empty

• ID3 leaf predictions
  
  • Most common labels (majority voting)

• ID3 split criteria
  
  • Information gain
ID3: Using Information Gain as Selection Criteria

- Information gain of choosing feature $A$ to split
  - $Gain(D, A) = H(D) - \sum_i \frac{|D_i|}{|D|} H(D_i)$
- ID3: Choose the split that maximize $Gain(D, A)$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>-1</td>
<td>+1</td>
</tr>
<tr>
<td>-1</td>
<td>+1</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

$H(D) = 0.5 \log_2 2 + 0.5 \log_2 2 = 1$

$H(D_{x_1=1}) = 0$ $H(D_{x_1=-1}) = 0$

$Gain(D, x_1) = 1$

$H(D_{x_2=1}) = 1$ $H(D_{x_2=-1}) = 1$

$Gain(D, x_2) = 0$

ID3 will choose $x_1$ as the next split attribute
Further Addressing Overfitting

• More Regularization (Constrain $H$)
  • Do not split leaves past a fixed depth
  • Do not split leaves with fewer than $c$ labels
  • Do not split leaves where the maximal information gain is less than $\tau$

• Pruning (removing leaves)
  • Evaluate each split using a validation set and compare the validation error with and without that split (replacing it with the most common label at that point)
  • Use statistical test to examine whether the split is “informative” (leads to different enough subtrees)
More Discussions

• Real-valued features (continuous $x$)
  • Need to select threshold for branching

• Regression (continuous $y$)
  • Change leaf prediction: e.g., average instead of majority vote
  • Change measure for “purity” of data: e.g., squared error of data