CSE 417T
Introduction to Machine Learning

Lecture 15
Instructor: Chien-Ju (CJ) Ho
Logistics: Homework 3

• Homework 3 is posted on the course website

• Due on March 25 (Wednesday), 2020
  • Homework 4 will be announced before the due of homework 3
Discussion on Exam 1
417T Part 2
Machine Learning Techniques
UNKNOWN TARGET DISTRIBUTION
(target function $f$ plus noise)
$P(y \mid x)$

$y_i \sim P(y \mid x_i)$

TRAINING EXAMPLES
$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$

UNKNOWN INPUT DISTRIBUTION
$P(x)$

$\mathcal{H}$

LEARNING ALGORITHM
$A$

ERROR MEASURE

FINAL HYPOTHESIS
$g$

$g(x) \approx f(x)$

$x_1, x_2, \ldots, x_N$
Focus of the rest of the semester
Decision Tree
Decision Tree Hypothesis

\[ \tilde{x} = (\text{annual income, have debt}) \]
\[ y \in \{\text{approve, deny}\} \]
Decision Tree Hypothesis

- **Pros**
  - Easy to interpret (interpretability is getting attention and is important in some 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

**Credit Card Approval Example**

```
Annual Income

≥ 100k                  ≥ 20k                  < 20k
    < 100k                  < 20k

Approve   have debt?   Deny

yes       no

Deny      Approve      Deny
```

Yes

No
**Decision Tree Hypothesis**

### Pros
- **Easy to interpret** (interpretability is getting attention and is important in some 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**

1. **Annual Income**
   - $\geq 100k$
   - $\geq 20k$
   - $< 20k$
   - $< 100k$

2. **have debt?**
   - **yes**
   - **no**

3. **Approve**
   - **Deny**

---

Annual Income $\geq 100k$
Annual Income $\geq 20k$
Annual Income $< 100k$
Annual Income $< 20k$

Deny
Deny
Approve
Approve

Credit Card Approval Example
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,

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

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

- **DecisionTreeLearn(D):** Input a dataset $D$, output a decision tree hypothesis
  - Create a root node $r$
  - 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 $r$

- 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>
</tbody>
</table>

DecisionTreeLearn($D$)
Create a root node $r$
If termination conditions are met
return a single node tree with leaf prediction based on
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 $r$

Termination conditions not met
Find a feature to split

Leaf prediction +1

Find next feature to split

Don’t terminate

terminate

+1

-1

$x_1$

$+1$ +1 +1 +1
-1 +1 -1 +1
+1 -1 +1 +1
-1 -1 -1 -1

$-1$ +1 +1 +1
-1 +1 -1 +1
-1 -1 +1 -1
-1 -1 -1 -1
Example Heuristics

• 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?

DecisionTreeLearn(\(D\))
Create a root node \(r\)
If termination conditions are met
  return a single node tree with leaf prediction based on
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 \(r\)
Split Criteria

• Which feature would you choose to split?

<table>
<thead>
<tr>
<th>x₁</th>
<th>x₂</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 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

• 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 => all labels are +1 or -1 => entropy = 0

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

By definition

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

- Information gain of choosing feature $A$ to split
  \[
  \text{Gain}(D, A) = H(D) - \sum_{i} \frac{|D_i|}{|D|} H(D_i) \quad \text{[The amount of decrease in entropy]}
  \]
  
- ID3: Choose the split that maximize $\text{Gain}(D, A)$

**DecisionTreeLearn($D$)**
- Create a root node $r$
- If termination conditions are met
  - return a single node tree with leaf prediction based on
- 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 $\text{DecisionTreeLearn}(D_i)$ that being the child of root $r$

**Notation:**
- $|D|$ is the number of points in $D$

**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 $Gain(D, A)$

\[
\begin{array}{ccc}
  x_1 & x_2 & y \\
  +1 & +1 & +1 \\
  +1 & -1 & +1 \\
  -1 & +1 & -1 \\
  -1 & -1 & -1 \\
\end{array}
\]

\[
H(D) = 0.5 \log_2 2 + 0.5 \log_2 2 = 1
\]

\[
\begin{align*}
  \text{Gain}(D, x_1) &= 1 \\
  \text{Gain}(D, x_2) &= 0
\end{align*}
\]

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
Ensemble Learning

The focus of the next two lectures
Ensemble Learning

• Assume we are given a set of learned hypothesis
  • $g_1, g_2, ..., g_M$

• What can we do?
  • Use validation to pick the best one
  • What if all of them are not good enough

• Can we aggregate them?
Is Aggregation a Good Idea?

• At a 1906 country fair, ~800 people participate in a contest to guess the weight of an ox.

• Reward is given to the person with the closest guess.

• The average guess is 1,197lbs. The true answer is 1,198lbs.
Is Aggregation a Good Idea?

• Maybe
  • If the hypothesis is “diverse”, and “in average” they seem good

• Question:
  • How do we find a set of hypothesis that are diverse and “in average” good
  • How do we aggregate the set of hypothesis

• Ensemble learning
  • Bagging – Random Forest (March 17)
  • Boosting – AdaBoost (March 19)