CSE/STAT 416
Naïve Bayes and Decision Trees
Pre-Class Videos

Hunter Schafer
Paul G. Allen School of Computer Science & Engineering
University of Washington

April 26, 2021
Idea: Estimate probabilities \( \hat{P}(y|x) \) and use those for prediction

**Probability Classifier**

Input \( x \): Sentence from review

- Estimate class probability \( \hat{P}(y = +1|x) \)
- If \( \hat{P}(y = +1|x) > 0.5 \):
  - \( \hat{y} = +1 \)
- Else:
  - \( \hat{y} = -1 \)

**Notes:**

- Estimating the probability improves *interpretability*
Interpreting Score

\[ Score(x_i) = w^T h(x_i) \]

\[ \hat{y}_i = \begin{cases} -1 & \text{Very sure} \\ 0 & \text{Not sure if} \\ +1 & \text{Very sure} \end{cases} \]

\[ \begin{align*}
\hat{P}(y_i = +1|x_i) &= 0 \\
\hat{P}(y_i = +1|x_i) &= 0.5 \\
\hat{P}(y_i = +1|x_i) &= 1 \\
\hat{P}(y = +1|x) &= 0.5 \\
\hat{P}(y = +1|x) &= 1
\end{align*} \]
\[ \hat{P}(y = +1|x, \hat{w}) = \text{sigmoid} \left( \hat{w}^T h(x) \right) = \frac{1}{1 + e^{-\hat{w}^T h(x)}} \]
Naïve Bayes
Idea: Naïve Bayes

\[ x = "The sushi & everything else was awesome!" \]

\[
P(y = +1 \mid x = "The sushi & everything else was awesome!")?
\]

\[
P(y = -1 \mid x = "The sushi & everything else was awesome!")?
\]

Idea: Select the class that is the most likely!

Bayes Rule:

\[
P(y = +1 \mid x) = \frac{P(x \mid y = +1)P(y = +1)}{P(x)}
\]

Example

\[
P("The sushi & everything else was awesome!" \mid y = +1) \frac{P(y = +1)}{P("The sushi & everything else was awesome!")}
\]

Since we’re just trying to find out which class has the greater probability, we can discard the divisor.
Naïve Assumption

**Idea:** Select the class with the highest probability!

**Problem:** We have not seen the sentence before.

**Assumption:** Words are independent from each other.

\[ x = \text{“The sushi & everything else was awesome!”} \]

\[
P(\text{“The sushi & everything else was awesome!”} | y = +1) \frac{P(y = +1)}{P(\text{“The sushi & everything else was awesome!”})} 
\]

\[
P(\text{“The sushi & everything else was awesome!”} | y = +1) = P(\text{The} | y = +1) \times P(\text{sushi} | y = +1) \times P(\& | y = +1) 
\times P(\text{everything} | y = +1) \times P(\text{else} | y = +1) \times P(\text{was} | y = +1) 
\times P(\text{awesome} | y = +1) 
\]
How do we compute something like

\[ P(y = +1) \]?

How do we compute something like

\[ P(\text{"awesome"} | y = +1) \]?
If a feature is missing in a class everything becomes zero.

\[
P(\text{“The sushi & everything else was awesome!”} | y = +1) \\
= P(\text{The} | y = +1) \times P(\text{sushi} | y = +1) \times P(\& | y = +1) \\
\times P(\text{everything} | y = +1) \times P(\text{else} | y = +1) \times P(\text{was} | y = +1) \\
\times P(\text{awesome} | y = +1)
\]

Solutions?

- Take the log (product becomes a sum).
  - Generally define \( \log(0) = 0 \) in these contexts
- Laplacian Smoothing (adding a constant to avoid multiplying by zero)
**Compare Models**

**Logistic Regression:**

\[ P(y = +1|x, w) = \frac{1}{1 + e^{-w^T h(x)}} \]

**Naïve Bayes:**

\[ P(y|x_1, x_2, ..., x_d) = \prod_{j=1}^{d} P(x_j|y) \cdot P(y) \]
Compare Models

**Generative:** defines a model for generating $x$ (e.g. Naïve Bayes)

**Discriminative:** only cares about defining and optimizing a decision boundary (e.g. Logistic Regression)
CSE/STAT 416
Naïve Bayes and Decision Trees

Hunter Schafer
Paul G. Allen School of Computer Science & Engineering
University of Washington

April 26, 2021

❓ Questions? Raise hand or sli.do #cs416
💬 Before Class: Pro-rain or anti-rain person?
🎵 Listening to: Alvvays
Logistic Regression:

\[
P(y = +1|x, w) = \frac{1}{1 + e^{-w^T h(x)}}
\]

Naïve Bayes:

\[
P(y|x_1, x_2, ..., x_d) = \prod_{j=1}^{d} P(x_j|y) \cdot P(y)
\]

- Based on counts of words/classes
  - Laplace Smoothing
Compare Models

**Generative:** defines a model for generating $x$ (e.g. Naïve Bayes)

**Discriminative:** only cares about defining and optimizing a decision boundary (e.g. Logistic Regression)
Recap: What is the predicted class for this sentence assuming we have the following training set (no Laplace Smoothing). “he is not cool”
Decision Trees
Humans often make decisions based on Flow Charts or Decision Trees.
### Parametric vs. Non-Parametric Methods

**Parametric Methods:** make assumptions about the data distribution

- Linear Regression ⇒ assume the data is linear
- Logistic Regression ⇒ assume probability has the shape of a logistic curve and linear decision boundary
- Those assumptions result in a **parameterized** function family. Our learning task is to learn the parameters.

**Non-Parametric Methods:** (mostly) don’t make assumptions about the data distribution

- Decision Trees, k-NN (soon)
- We’re still learning something, but not the parameters to a function family that we’re assuming describes the data.
- Useful when you don’t want to (or can’t) make assumptions about the data distribution.
- A line might not always support our decisions.
What makes a loan risky?

I want to buy a new house!

- Credit History: ★★★★★
- Income: ★★★
- Term: ★★★★★★
- Personal Info: ★★★

Loan Application
Credit history explained

Did I pay previous loans on time?

Example: excellent, good, or fair

Credit History ★★★★★

Income ★★★

Term ★★★★★★

Personal Info ★★★
What’s my income?
Example:
$80K per year
How soon do I need to pay the loan?

Example: 3 years, 5 years,...
Personal information

Age, reason for the loan, marital status,...

Example: Home loan for a married couple
Intelligent application
Classifier review

Input: $x_i$

Classifier MODEL

Output: $\hat{y}$
Predicted class

$\hat{y}_i = +1$  Safe
$\hat{y}_i = -1$  Risky
Setup

Data (N observations, 3 features)

<table>
<thead>
<tr>
<th>Credit</th>
<th>Term</th>
<th>Income</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>excellent</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>5 yrs</td>
<td>low</td>
<td>risky</td>
</tr>
<tr>
<td>fair</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
<tr>
<td>poor</td>
<td>5 yrs</td>
<td>high</td>
<td>risky</td>
</tr>
<tr>
<td>excellent</td>
<td>3 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>5 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>poor</td>
<td>3 yrs</td>
<td>high</td>
<td>risky</td>
</tr>
<tr>
<td>poor</td>
<td>5 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
</tbody>
</table>

Evaluation: classification error

Many possible decisions: number of trees grows exponentially!
With our discussion of bias and fairness from last week, discuss the potential biases and fairness concerns that might be present in our dataset about loan safety.
Decision Trees

- **Branch/Internal node:** splits into possible values of a feature
- **Leaf node:** final decision (the class value)
Loan status: Safe  Risky

Root
6  3

# of Safe loans

# of Risky loans

N = 9 examples
Decision stump: 1 level

Loan status: Safe Risky

<table>
<thead>
<tr>
<th>Credit</th>
<th>Term</th>
<th>Income</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>excellent</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>5 yrs</td>
<td>low</td>
<td>risky</td>
</tr>
<tr>
<td>fair</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
<tr>
<td>poor</td>
<td>5 yrs</td>
<td>high</td>
<td>risky</td>
</tr>
<tr>
<td>excellent</td>
<td>3 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>5 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>poor</td>
<td>3 yrs</td>
<td>high</td>
<td>risky</td>
</tr>
<tr>
<td>poor</td>
<td>5 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
</tbody>
</table>

Split on Credit

Subset of data with Credit = excellent

Subset of data with Credit = fair

Subset of data with Credit = poor
Making predictions

For each leaf node, set $\hat{y} = \text{majority value}$

Loan status: Safe  Risky

credit?

excellent 2 0

fair 3 1

poor 1 2

Safe  Safe  Risky
How do we select the best feature?

- Select the split with lowest classification error

**Choice 1: Split on Credit**

- Loan status: Safe Risky
- Root 6 3
- Credit?
- excellent 2 0
- fair 3 1
- poor 1 2

**Choice 2: Split on Term**

- Loan status: Safe Risky
- Root 6 3
- Term?
- 3 years 4 1
- 5 years 2 1
Calculate the node values.

<table>
<thead>
<tr>
<th>Credit</th>
<th>Term</th>
<th>Income</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>excellent</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>5 yrs</td>
<td>low</td>
<td>risky</td>
</tr>
<tr>
<td>fair</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
<tr>
<td>poor</td>
<td>5 yrs</td>
<td>high</td>
<td>risky</td>
</tr>
<tr>
<td>excellent</td>
<td>3 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>5 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>poor</td>
<td>3 yrs</td>
<td>high</td>
<td>risky</td>
</tr>
<tr>
<td>poor</td>
<td>5 yrs</td>
<td>low</td>
<td>safe</td>
</tr>
<tr>
<td>fair</td>
<td>3 yrs</td>
<td>high</td>
<td>safe</td>
</tr>
</tbody>
</table>

Choice 2: Split on Term

Loan status: Safe Risky

Root

6 3

Term?

3 years

5 years
How do we select the best feature?

Select the split with lowest classification error

**Choice 1: Split on Credit**

- **Loan status:** Safe  Risky
- **Root:** 6  3
  - **Credit?**
    - excellent: 2  0
    - fair: 3  1
    - poor: 1  2

**Choice 2: Split on Term**

- **Loan status:** Safe  Risky
- **Root:** 6  3
  - **Term?**
    - 3 years: 4  1
    - 5 years: 2  2
How do we measure effectiveness of a split?

Idea: Calculate classification error of this decision stump

Error = \frac{\text{# mistakes}}{\text{# data points}}
Calculating classification error

Step 1: \( \hat{y} = \) class of majority of data in node
Step 2: Calculate classification error of predicting \( \hat{y} \) for this data

\[ \text{Error} = \frac{3 \text{ mistakes}}{6 \text{ correct}} = 0.5 \]

<table>
<thead>
<tr>
<th>Tree</th>
<th>Classification error</th>
</tr>
</thead>
<tbody>
<tr>
<td>(root)</td>
<td>0.33</td>
</tr>
</tbody>
</table>
Does a split on Credit reduce classification error below 0.33?
Choice 1: Split on Credit

Loan status:
Safe  Risky

Root
6 3

Credit?

excellent
2 0
Safe
0 mistakes

fair
3 1
Safe
1 mistake

poor
1 2
Risky
1 mistake

Error = ______

Tree | Classification error
--- | ---
(root) | 0.33
Split on credit | 0.22
Choice 2: Split on Term

Loan status:
Safe  Risky

Root
6  3

Term?

3 years
4  1
Safe

5 years
2  2
Risky
Evaluating the split on Term

Choice 2: Split on Term

Loan status: Safe Risky

Root

3 years
4 1

5 years
2 2

Term?

Error = ______

Tree | Classification error
---|---
(root) | 0.33
Split on credit | 0.22
Split on term | 0.33
Choice 1 vs Choice 2: Comparing split on credit vs term

<table>
<thead>
<tr>
<th>Tree</th>
<th>Classification error</th>
</tr>
</thead>
<tbody>
<tr>
<td>(root)</td>
<td>0.33</td>
</tr>
<tr>
<td>split on credit</td>
<td>0.22</td>
</tr>
<tr>
<td>split on loan term</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Choice 1: Split on Credit

Choice 2: Split on Term
Split Selection

Split(node)

- Given $M$, the subset of training data at a node
- For each (remaining) feature $h_j(x)$:
  - Split data $M$ on feature $h_j(x)$
  - Compute the classification error for the split
- Chose feature $h_j^*(x)$ with the lowest classification error
Greedy & Recursive Algorithm

**BuildTree(node)**

- If termination criterion is met:
  - Stop
- Else:
  - Split(node)
  - For child in node:
    - BuildTree(child)
Decision stump: 1 level

Loan status: Safe Risky

Split on Credit

- **Root**
  - 6 Safe
  - 3 Risky

- **Credit?**
  - excellent
    - Subset of data with Credit = excellent
    - 2 Safe
    - 0 Risky
  - fair
    - Subset of data with Credit = fair
    - 3 Safe
    - 1 Risky
  - poor
    - Subset of data with Credit = poor
    - 1 Safe
    - 2 Risky
For now: Stop when all points are in one class

Loan status:
Safe Risky

Root
6 3

Credit?

excellent
2 0

fair
3 1

poor
2 1

Safe

All data points are Safe nothing else to do with this subset of data

Leaf node
Tree learning = Recursive stump learning

Loan status: Safe Risky

Build decision stump with subset of data where Credit = fair

Build decision stump with subset of data where Credit = poor
What predictions **should** the below decision tree output for the following datapoints?

**Loan status:**
- Safe
- Risky

### Decision Tree

- **Root:**
  - Credit?
    - Excellent: 2 Safe, 0 Risky
    - Fair: 3 Safe, 1 Risky
    - Poor: 1 Risky, 2 Safe

### Data Table

<table>
<thead>
<tr>
<th>Credit</th>
<th>Term</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excellent</td>
<td>5 yrs</td>
<td>high</td>
</tr>
<tr>
<td>Fair</td>
<td>3 yrs</td>
<td>low</td>
</tr>
<tr>
<td>Poor</td>
<td>5 yrs</td>
<td>(missing)</td>
</tr>
</tbody>
</table>
What predictions **should** the below decision tree output for the following datapoints?

<table>
<thead>
<tr>
<th>Credit</th>
<th>Term</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>excellent</td>
<td>5 yrs</td>
<td>high</td>
</tr>
<tr>
<td>fair</td>
<td>3 yrs</td>
<td>low</td>
</tr>
<tr>
<td>poor</td>
<td>5 yrs</td>
<td>(missing)</td>
</tr>
</tbody>
</table>

**Loan status:**

Safe  Risky
Brain Break
Real valued features

<table>
<thead>
<tr>
<th>Income</th>
<th>Credit</th>
<th>Term</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>$105 K</td>
<td>excellent</td>
<td>3 yrs</td>
<td>Safe</td>
</tr>
<tr>
<td>$112 K</td>
<td>good</td>
<td>5 yrs</td>
<td>Risky</td>
</tr>
<tr>
<td>$73 K</td>
<td>fair</td>
<td>3 yrs</td>
<td>Safe</td>
</tr>
<tr>
<td>$69 K</td>
<td>excellent</td>
<td>5 yrs</td>
<td>Safe</td>
</tr>
<tr>
<td>$217 K</td>
<td>excellent</td>
<td>3 yrs</td>
<td>Risky</td>
</tr>
<tr>
<td>$120 K</td>
<td>good</td>
<td>5 yrs</td>
<td>Safe</td>
</tr>
<tr>
<td>$64 K</td>
<td>fair</td>
<td>3 yrs</td>
<td>Risky</td>
</tr>
<tr>
<td>$340 K</td>
<td>excellent</td>
<td>5 yrs</td>
<td>Safe</td>
</tr>
<tr>
<td>$60 K</td>
<td>good</td>
<td>3 yrs</td>
<td>Risky</td>
</tr>
</tbody>
</table>
Threshold split

Loan status:
Safe Risky

Split on Income

Income?

Subset of data with Income

>= $60K

< $60K

8 13

14 5
Best threshold?

Similar to our simple, threshold model when discussing Fairness!

Infinite possible values of $t$

Income = $t^*$

Income < $t^*$

Income ≥ $t^*$

Safe

Risky

Income

$10K

$120K
Threshold between points

Same classification error for any threshold split between \( v_a \) and \( v_b \)
Only need to consider mid-points

Finite number of splits to consider
Threshold split selection algorithm

- **Step 1:** Sort the values of a feature $h_j(x)$:
  Let $[v_1, v_2, ..., v_N]$ denote sorted values

- **Step 2:**
  - For $i = [1, ..., N - 1]$
    - Consider split $t_i = \frac{v_i + v_{i+1}}{2}$
    - Compute classification error for threshold split $h_j(x) \geq t_i$
  - Chose the $t^*$ with the lowest class. error
Visualizing the threshold split

Threshold split is the line $\text{Age} = 38$
Split on Age $\geq$ 38

- Predict Safe
- Predict Risky

- Age
- Income

- $0K$, $10K$, $20K$, $30K$, $40K$, $80K$
Each split partitions the 2-D space.
Depth 1: Split on $x[1]$
Threshold split caveat

For threshold splits, same feature can be used multiple times
Decision boundaries

- Decision boundaries can be complex!
Overfitting

- Deep decision trees are prone to overfitting
  - Decision boundaries are interpretable but not stable
  - Small change in the dataset leads to big difference in the outcome

- Overcoming Overfitting:
  - Stop when tree reaches certain height (e.g., 4 levels)
  - Stop when leaf has $\leq$ some num of points (e.g., 20 pts)
    - Will be the stopping condition for HW
  - Stop if split won’t significantly decrease error by more than some amount (e.g., 10%)

- Other methods include growing full tree and pruning back

- Fine-tune hyperparameters with validation set or CV
Trees can be used for classification or regression (CART)
- Classification: Predict majority class for root node
- Regression: Predict average label for root node

In practice, we don’t minimize classification error but instead some more complex metric to measure quality of split such as Gini Impurity or Information Gain (not covered in 416)

Can also be used to predict probabilities
Predicting probabilities

Loan status: Safe Risky

P(y = Safe | x) = \frac{3}{3 + 1} = 0.75
Recap

What you can do now:

- Define the assumptions and modeling for Naïve Bayes
- Define a decision tree classifier
- Interpret the output of a decision trees
- Learn a decision tree classifier using greedy algorithm
- Traverse a decision tree to make predictions
  - Majority class predictions