23: Naïve Bayes

Jerry Cain
May 22, 2024

Lecture Discussion on Ed
Preamble: Machine Learning
The Path Before Us

Parameter Estimation

- Deep Learning
- Linear Regression
- Naïve Bayes
- Logistic Regression
The Path Before Us

- Linear Regression
- Naïve Bayes
- Logistic Regression
- Deep Learning

Unbiased estimators: \( \bar{X}, S^2 \)  
Maximizing likelihood: \( \theta_{\text{MLE}} \)  
Bayesian estimation: \( \theta_{\text{MAP}} \)

MLE: choosing parameter that maximizes probability of seeing the observed data.

MAP: choose most likely \( \theta \) value given prior and experimental data.
Machine Learning uses a lot of data.

Many different forms of machine learning
- We focus on the problem of prediction given prior observations.

Task: Identify the chair
Data: All the chairs ever

Supervised learning: A category of machine learning where you have labeled data for the problem you are solving.
Supervised learning

Real World Problem

Model the problem

Formal Model $\theta$

Learning Algorithm

Testing Data

Prediction Function $\hat{\theta}$

Training Data

Evaluation score
Supervised learning

Modeling

Not CS109’s focus. **CS228** is awesome.
Supervised learning

Parameter estimation is a basis for learning from data.

Real World Problem

Model the problem

Formal Model $\theta$

Learning Algorithm

Testing Data

Prediction Function $\hat{\theta}$

Evaluation score

Training Data

Lisa Yan, Chris Piech, Monica Ojha, and Jerry Cain, CS109, Spring 2024
Model and dataset

Many different forms of machine learning
• We focus on one specific type: prediction from past observations.

Goal
   Based on observed $X$, predict some unknown $Y$

• Features
   Vector $X$ of $m$ observations (new term: feature vector)
   $X = (X_1, X_2, ..., X_m)$

• Output
   Variable $Y$ (also called class label if discrete)

Model
   $\hat{Y} = g(X)$, a function on $X$

Examples:
- Dental X-ray $\rightarrow$ predict cavities
- Most recently typed character $\rightarrow$ predict word being typed
- Temperatures $\rightarrow$ predict tomorrow’s weather
## Training data

\[
X = (X_1, X_2, X_3, ..., X_{300})
\]

We assume all input features are Bernoulli random variables.

<table>
<thead>
<tr>
<th>Feature 1</th>
<th>Feature 2</th>
<th>Feature 300</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient 1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Patient 2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Patient n</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Spring 2024
Training data notation

\[(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})\]

\(n\) datapoints, assumed to be iid

\(i\)-th datapoint \((x^{(i)}, y^{(i)})\):

- \(m\) features: \(x^{(i)} = (x_{1}^{(i)}, x_{2}^{(i)}, \ldots, x_{m}^{(i)})\)
- A single output \(y^{(i)}\)
- Independent of all other datapoints

Training Goal: Use these \(n\) datapoints to learn a model \(\hat{Y} = g(X)\) that predicts \(Y\)
Supervised learning

Training: use both $x$'s and $y$'s to train the model—that is, estimate the parameters.

Testing: feed different set of $x$'s to fully trained model to see how well predicted matches known $y$ values, i.e. does $\hat{y} = y$?

Real World Problem

Model the problem

Formal Model $\theta$

Learning Algorithm

Training Data

Testing Data

Prediction Function $\hat{\theta}$

Evaluation score
Testing data notation

\[(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(n)}, y^{(n)})\]

\(n_{test}\) other datapoints, assumed to be iid

\(i^{th}\) datapoint \((x^{(i)}, y^{(i)})\):
- Has the same structure as your training data

Testing Goal: Leveraging the model \(\hat{Y} = g(X)\) that you trained, see how well you can predict \(Y\) on known data
Two tasks we will focus on

Many different forms of machine learning
• We focus on the problem of **prediction** based on observations.

Goal
• Based on observed $X$, predict some unknown $Y$

Features
• Vector $X$ of $m$ observations (new term: feature vector)
  $X = (X_1, X_2, \ldots, X_m)$

Output
• Variable $Y$ (also called **class label** if discrete)

Model
• $\hat{Y} = g(X)$, a function on $X$
• **Regression** prediction when $Y$ is continuous
• **Classification** prediction when $Y$ is discrete

*today's focus*
Regression: Predicting real numbers

Training data: \((x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})\)

- CO2 levels
- Sea level
- Feature \(m\)
- Output

<table>
<thead>
<tr>
<th>Year</th>
<th>CO2 levels</th>
<th>Sea level</th>
<th>Feature (m)</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year 1</td>
<td>338.8</td>
<td>0</td>
<td>1</td>
<td>0.26</td>
</tr>
<tr>
<td>Year 2</td>
<td>340.0</td>
<td>1</td>
<td>0</td>
<td>0.32</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year (n)</td>
<td>340.76</td>
<td>0</td>
<td>1</td>
<td>0.14</td>
</tr>
</tbody>
</table>

_next! We’ll study this a bit next lecture_
**Classification:** Predicting class labels

\[ X = (X_1, X_2, X_3, \ldots, X_{300}) \]

<table>
<thead>
<tr>
<th>Patient</th>
<th>Feature 1</th>
<th>Feature 2</th>
<th>Feature 300</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>n</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
</tr>
</tbody>
</table>

outputs (aka labels) aren't always binary, but they are in today's examples.
Classification: Healthy hearts

$X = (X_1)$

Feature 1 | Output
---|---
Patient 1 | 0
Patient 2 | 1
... | ...
Patient $n$ | 1

Single feature: Region of Interest (ROI) is healthy (1) or unhealthy (0)

How can we predict whether heart is healthy (1) or not (0)?

The following strategy is **not used in practice** but helps us understand how to approach classification.
Classification: Brute Force Bayes

\[ \hat{Y} = g(X) \]

Our prediction for \( Y \) is a function of \( X \)

\[ \begin{align*}
\hat{Y} &= \arg \max_{y \in \{0,1\}} P(Y | X) \\
&= \arg \max_{y \in \{0,1\}} \frac{P(X|Y)P(Y)}{P(X)} \\
&= \arg \max_{y \in \{0,1\}} P(X|Y)P(Y)
\end{align*} \]

Proposed model: Choose the \( Y \) that is more or most likely given \( X \)

\( \frac{P(X|Y)}{P(X)} \) (Bayes’ Theorem)

\( 1/P(X) \) is constant wrt \( y \)

If we estimate \( P(X|Y) \) and \( P(Y) \), we can classify data points.
Training: Estimate parameters

\[ X = (X_1) \]

<table>
<thead>
<tr>
<th>Feature 1</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient 1</td>
<td>1</td>
</tr>
<tr>
<td>Patient 2</td>
<td>1</td>
</tr>
<tr>
<td>Patient n</td>
<td>0</td>
</tr>
</tbody>
</table>

Conditional probability tables \( \hat{P}(X|Y) \)

\[
\begin{array}{c|c|c}
X_1 & \hat{P}(X|Y = 0) & \hat{P}(X|Y = 1) \\
\hline
0 & \theta_1 & \theta_3 \\
1 & \theta_2 = \theta_1 & \theta_4 = \theta_3 \\
\end{array}
\]

Marginal probability table \( \hat{P}(Y) \)

\[
\begin{array}{c|c|c}
Y & \hat{P}(Y) & \hat{P}(Y)
\hline
0 & \theta_5 & \\
1 & \theta_6 = \theta_5 & \\
\end{array}
\]

\( \hat{Y} \) = \arg \max_{y \in \{0,1\}} \hat{P}(X|Y)\hat{P}(Y)

Training Goal: Use \( n \) datapoints to learn \( 2 \cdot 2 + 2 = 6 \) parameters.
Training: Estimate parameters $\hat{P}(X|Y)$

<table>
<thead>
<tr>
<th>Count:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1 = 0, Y = 0$: 4</td>
</tr>
<tr>
<td>$X_1 = 1, Y = 0$: 6</td>
</tr>
<tr>
<td>$X_1 = 0, Y = 1$: 0</td>
</tr>
<tr>
<td>$X_1 = 1, Y = 1$: 100</td>
</tr>
<tr>
<td>Total: 110</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Patient $\ n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

| $\hat{P}(X|Y = 0)$ |
|---------------------|
| $X_1 = 0$: $\theta_1$ |
| $X_1 = 1$: $\theta_2 = 1 - \theta_1$ |

| $\hat{P}(X|Y = 1)$ |
|---------------------|
| $\theta_3$ |
| $\theta_4 = 1 - \theta_3$ |

Use MLE or Laplace (MAP) estimate $\hat{P}(X|Y)$ and $\hat{P}(Y)$ as parameters.
Training: MLE estimates, $\hat{P}(X|Y)$

|          | $\hat{P}(X|Y = 0)$ | $\hat{P}(X|Y = 1)$ |
|----------|----------------------|----------------------|
| $X_1 = 0$| 0.4 $= \frac{4}{4+6}$ | 0.0 $= \frac{0}{12}$ |
| $X_1 = 1$| 0.6 $= \frac{6}{4+6}$ | 1.0 $= \frac{10}{15}$ |

MLE of $\hat{P}(X_1 = x|Y = y) = \frac{#(X_1 = x, Y = y)}{#(Y = y)}$

Just count!

<table>
<thead>
<tr>
<th>Count:</th>
<th># datanoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1 = 0, Y = 0$:</td>
<td>4</td>
</tr>
<tr>
<td>$X_1 = 1, Y = 0$:</td>
<td>6</td>
</tr>
<tr>
<td>$X_1 = 0, Y = 1$:</td>
<td>0</td>
</tr>
<tr>
<td>$X_1 = 1, Y = 1$:</td>
<td>100</td>
</tr>
<tr>
<td>Total:</td>
<td>110</td>
</tr>
</tbody>
</table>
### Training: Laplace (MAP) estimates, $\hat{P}(X|Y)$

|        | $\hat{P}(X|Y = 0)$ | $\hat{P}(X|Y = 1)$ |
|--------|--------------------|--------------------|
| $X_1 = 0$ | 0.4                | 0.0                |
| $X_1 = 1$ | 0.6                | 1.0                |

**MLE of $\hat{P}(X_1 = x|Y = y)$**

Just count!

$$\hat{P}(X_1 = x|Y = y) = \frac{\#(X_1 = x, Y = y)}{\#(Y = y)}$$

#### Example

$$P(X_1 = 1|Y = 0) = \frac{\#(X_1 = 1, Y = 0)+1}{\#(Y = 0)+1} = \frac{6+1}{10+2} = \frac{7}{12}$$

Laplace of $\hat{P}(X_1 = x|Y = y) = ?$

Just count + add imaginary trials!
Training: Laplace (MAP) estimates, $\hat{P}(X|Y)$

| $X_1$ | $\hat{P}(X|Y = 0)$ | $\hat{P}(X|Y = 1)$ |
|-------|----------------------|----------------------|
| $X_1 = 0$ | 0.4 | 0.0 |
| $X_1 = 1$ | 0.6 | 1.0 |

MLE of $\hat{P}(X_1 = x|Y = y) = \frac{\#(X_1 = x,Y = y)}{\#(Y = y)}$

Just count!

| $X_1$ | $\hat{P}(X|Y = 0)$ | $\hat{P}(X|Y = 1)$ |
|-------|----------------------|----------------------|
| $X_1 = 0$ | $0.42 = \frac{5}{12}$ | $0.01 = \frac{1}{102}$ |
| $X_1 = 1$ | $0.58 = \frac{7}{12}$ | $0.99 = \frac{101}{102}$ |

Laplace of $\hat{P}(X_1 = x|Y = y) = \frac{\#(X_1 = x,Y = y) + 1}{\#(Y = y) + 2}$

Just count + add imaginary trials!
Testing

\[ \hat{Y} = \arg \max_{Y \in \{0,1\}} \hat{P}(X|Y)\hat{P}(Y) \]

| (MAP) | \( \hat{P}(X|Y = 0) \) | \( \hat{P}(X|Y = 1) \) |
|-------|-----------------|-----------------|
| \( X_1 = 0 \) | 0.42 | 0.01 |
| \( X_1 = 1 \) | 0.58 | 0.99 |

<table>
<thead>
<tr>
<th>(MLE)</th>
<th>( \hat{P}(Y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y = 0 )</td>
<td>0.09</td>
</tr>
<tr>
<td>( Y = 1 )</td>
<td>0.91</td>
</tr>
</tbody>
</table>

New patient has a healthy ROI (\( X_1 = 1 \)). What is your prediction, \( \hat{Y} \)?

\[
\hat{P}(X_1 = 1|Y = 0)\hat{P}(Y = 0) = 0.58 \cdot 0.09 \approx 0.052 \\
\hat{P}(X_1 = 1|Y = 1)\hat{P}(Y = 1) = 0.99 \cdot 0.91 \approx 0.901
\]

A. \( 0.052 < 0.5 \) \( \Rightarrow \hat{Y} = 1 \)

B. \( 0.901 > 0.5 \) \( \Rightarrow \hat{Y} = 1 \)

C. \( 0.052 < 0.901 \Rightarrow \hat{Y} = 1 \)
Brute Force Bayes classifier

\[ \hat{Y} = \arg \max_{y \in \{0, 1\}} \hat{P}(X|Y) \hat{P}(Y) \]

(\( \hat{P}(Y) \) is an estimate of \( P(Y) \),
\( \hat{P}(X|Y) \) is an estimate of \( P(X|Y) \))

Estimate these probabilities—i.e.,
learn these parameters using MLE or Laplace (MAP)

\[
\begin{align*}
\hat{P}(X_1, X_2, \ldots, X_m | Y = 1) \\
\hat{P}(X_1, X_2, \ldots, X_m | Y = 0) \\
\hat{P}(Y = 1) & \quad \hat{P}(Y = 0)
\end{align*}
\]

Given an observation \( X = (X_1, X_2, \ldots, X_m) \), predict

\[ \hat{Y} = \arg \max_{y \in \{0, 1\}} \left( \hat{P}(X_1, X_2, \ldots, X_m | Y) \hat{P}(Y) \right) \]
Naïve Bayes
Brute Force Bayes: \( m = 300 \) (# features)

\[
X = (X_1, X_2, X_3, ..., X_{300})
\]

This won’t be too bad, right?
Brute Force Bayes: $m = 300$ (# features)

\[ X = (X_1, X_2, X_3, ..., X_{300}) \]

<table>
<thead>
<tr>
<th>Patient</th>
<th>$X_1$ = 0, $X_2$ = 0, ..., $X_{299}$ = 0, $X_{300}$ = 0, $Y$ = 0:</th>
<th># datapoints</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$X_1$ = 0, $X_2$ = 0, ..., $X_{299}$ = 0, $X_{300}$ = 1, $Y$ = 0:</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$X_1$ = 0, $X_2$ = 0, ..., $X_{299}$ = 1, $X_{300}$ = 0, $Y$ = 0:</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Patient</th>
<th>$X_1$ = 0, $X_2$ = 0, ..., $X_{299}$ = 0, $X_{300}$ = 1, $Y$ = 1:</th>
<th># datapoints</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$X_1$ = 0, $X_2$ = 0, ..., $X_{299}$ = 0, $X_{300}$ = 1, $Y$ = 1:</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$X_1$ = 0, $X_2$ = 0, ..., $X_{299}$ = 1, $X_{300}$ = 0, $Y$ = 1:</td>
<td>1</td>
</tr>
</tbody>
</table>

This won’t be too bad, right?
Brute Force Bayes: $m = 300$ (# features)

\[ \hat{Y} = \arg \max_{y \in \{0,1\}} \hat{P}(Y | X) \]
\[ = \arg \max_{y \in \{0,1\}} \frac{\hat{P}(X|Y)\hat{P}(Y)}{\hat{P}(X)} \]
\[ = \arg \max_{y \in \{0,1\}} \hat{P}(X|Y)\hat{P}(Y) \]

- $\hat{P}(Y = 1 | x)$: estimated probability a heart is healthy given $x$
- $X = (X_1, X_2, \ldots, X_{300})$: whether 300 regions of interest (ROI) are healthy (1) or unhealthy (0)

How many parameters do we have to learn?

- $\hat{P}(X|Y)$
- $\hat{P}(Y)$

A. $2 \cdot 2 + 2 = 6$
B. $2 \cdot 300 + 2 = 602$
C. $2 \cdot 2^{300} + 2 = \text{a lot}$

Learn parameters through MLE or MAP
Brute Force Bayes: \( m = 300 \) (# features)

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \hat{P}(Y | X) \\
= \arg \max_{y=\{0,1\}} \frac{\hat{P}(X|Y) \hat{P}(Y)}{\hat{P}(X)} \\
= \arg \max_{y=\{0,1\}} \hat{P}(X|Y) \hat{P}(Y)
\]

Learn parameters through MLE or MAP

This approach requires you to learn \( O(2^m) \) parameters.

- \( \hat{P}(Y = 1 | x) \): estimated probability a heart is healthy given \( x \)
- \( X = (X_1, X_2, ..., X_{300}) \): whether 300 regions of interest (ROI) are healthy (1) or unhealthy (0)

How many parameters do we have to learn?

\[
\hat{P}(X|Y) \quad \hat{P}(Y) \\
A. \quad 2 \cdot 2 + 2 = 6 \\
B. \quad 2 \cdot 300 + 2 = 602 \\
C. \quad 2 \cdot 2^{300} + 2 = a lot
\]

Each of 300 features can be either 0 or 1, \( \Rightarrow 2^{300} \), for each of \( Y = 0 \) and \( Y = 1 \).
The problem with our current classifier

$$\hat{Y} = \arg \max_{y=\{0,1\}} \hat{P}(Y \mid X)$$

Choose the $Y$ that is most likely given $X$

$$= \arg \max_{y=\{0,1\}} \frac{\hat{P}(X \mid Y)\hat{P}(Y)}{\hat{P}(X)}$$

(Bayes’ Theorem)

$$= \arg \max_{y=\{0,1\}} \hat{P}(X \mid Y)\hat{P}(Y)$$

(1/$P(X)$ is constant w.r.t. $y$)

$$\hat{P}(X_1, X_2, \ldots, X_m \mid Y)$$

Estimating this joint conditional distribution is intractable.

What if we could make a simplifying assumption—even if incredibly naïve—to make our parameter estimation effort computationally tractable?
The Naïve Bayes assumption

\[ \hat{Y} = \arg \max_{y\in\{0,1\}} \hat{P}(Y | X) \]

\[ = \arg \max_{y\in\{0,1\}} \frac{\hat{P}(X|Y)\hat{P}(Y)}{\hat{P}(X)} \]

\[ = \arg \max_{y\in\{0,1\}} \hat{P}(X|Y)\hat{P}(Y) \]

\[ = \arg \max_{y\in\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right)\hat{P}(Y) \]

Assumption:

\[ \hat{P}(X_1,X_2,...,X_m | Y) = \prod_{i=1}^{m} \hat{P}(X_i | Y) \]

\[ X_1, ..., X_m \text{ are all conditionally independent given } Y. \]

Naïve Bayes Assumption: by making this assumption, the number of parameters to compute becomes linear in \( m \) instead of exponential.
Naïve Bayes Classifier

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)
\]

**Training**

What is the Big-O of # of parameters we need to learn?

A. \( O(m) \)

B. \( O(2^m) \)

C. other
Naïve Bayes Classifier

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)
\]

Training

\[
\hat{P}(Y = 1) = \hat{P}(Y = 0) = 1 - \hat{P}(Y = 1)
\]

for \( j = 1, \ldots, m \):

\[
\hat{P}(X_j = 1|Y = 0), \quad \hat{P}(X_j = 1|Y = 1)
\]

Use MLE or Laplace (MAP)

\( \# \text{parameters to be computed is } 4m + 2 = \Theta(m) \)

Testing

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \log \hat{P}(Y) + \sum_{j=1}^{m} \log \hat{P}(X_j|Y) \right)
\]
NETFLIX

and Learn
Classification terminology check

Training data: \((x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})\)

User 1
1. \(x^{(1)} \Rightarrow A\)
2. \(y^{(1)} \Rightarrow B\)
3. \((x^{(1)}, y^{(1)}) \Rightarrow C\)
4. \(x^{(1)} \Rightarrow D\)

User 2
3. \(x^{(2)} \Rightarrow A\)
4. \(y^{(2)} \Rightarrow B\)
5. \((x^{(2)}, y^{(2)}) \Rightarrow C\)

User n
4. \(x^{(n)} \Rightarrow A\)
5. \(y^{(n)} \Rightarrow B\)
6. \((x^{(n)}, y^{(n)}) \Rightarrow C\)
Predicting user TV preferences

Will a user like the Pokémon TV series?

Observe indicator variables $X = (X_1, X_2)$:

- $X_1 = 1$: "likes Star Wars"
- $X_2 = 1$: "likes Harry Potter"

Output $Y$ indicator:

- $Y = 1$: "likes Pokémon"

Predict $\hat{Y} = \arg \max_{Y \in \{0,1\}} \hat{P}(Y | X)$

\[ \hat{Y} = \arg \max_{Y \in \{0,1\}} \hat{P}(Y | X) \]
Predicting user TV preferences

Which probabilities do you need to estimate? How many are there?

- Brute Force Bayes (strawman, without NB assumption)

- Naïve Bayes

During training, how to estimate the prob $\hat{P}(X_1 = 1, X_2 = 1 | Y = 0)$ with MLE? with Laplace?

- Brute Force Bayes

- Naïve Bayes
Ex 1. Naïve Bayes Classifier (MLE)

Training:

\[ \hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y) \]

∀i: \( \hat{P}(X_j = 1|Y = 0), \hat{P}(X_j = 0|Y = 0) \), Use MLE or Laplace (MAP)
\( \hat{P}(X_j = 1|Y = 1), \hat{P}(X_j = 0|Y = 0) \)
\( \hat{P}(Y = 1), \hat{P}(Y = 0) \)

Testing:

\[ \hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y) \]
Training: Naïve Bayes for TV shows (MLE)

Observe indicator vars. \( \mathbf{X} = (X_1, X_2) \):
- \( X_1 \): “likes Star Wars”
- \( X_2 \): “likes Harry Potter”

Predict \( Y \): “likes Pokémon”

1. How many datapoints (\( n \)) are in our training data?
2. Compute MLE estimates for \( \hat{P}(X_1|Y) \):

<table>
<thead>
<tr>
<th>( X_1 )</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y )</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( X_2 )</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y )</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

Training data counts

\[
\begin{align*}
\hat{P}(X_1 = 0|Y = 0) &= \frac{3}{13} \\
\hat{P}(X_1 = 1|Y = 0) &= \frac{10}{13} \\
\hat{P}(X_1 = 0|Y = 1) &= \frac{7}{10} \\
\hat{P}(X_1 = 1|Y = 1) &= \frac{3}{10}
\end{align*}
\]
Training: Naïve Bayes for TV shows (MLE)

Observe indicator vars. \( X = (X_1, X_2) \):
- \( X_1 \): “likes Star Wars”
- \( X_2 \): “likes Harry Potter”

Predict \( Y \): “likes Pokémon”

1. How many datapoints (\( n \)) are in our training data?

\[ n = 12 + 17 = 30 \]

2. Compute MLE estimates for \( \hat{P}(X_1|Y) \):

\[
\begin{array}{c|cc}
Y & 0 & 1 \\
\hline
1 & 10 & 13 \\
0 & 3 & 5 \\
\end{array}
\begin{array}{c|cc}
X_1 & 0 & 1 \\
\hline
1 & 7 & 10 \\
0 & 4 & 8 \\
\end{array}
\]

\[
\hat{P}(X_1 = 0 | Y = 0) = \frac{3}{13} \approx 0.23 \\
\hat{P}(X_1 = 0 | Y = 1) = \frac{4}{17} \approx 0.24 \\
\hat{P}(X_1 = 1 | Y = 0) = \frac{10}{13} \approx 0.77 \\
\hat{P}(X_1 = 1 | Y = 1) = \frac{13}{17} \approx 0.77
\]
## Training: Naïve Bayes for TV shows (MLE)

Observe indicator vars. $X = (X_1, X_2)$:
- $X_1$: “likes Star Wars”
- $X_2$: “likes Harry Potter”

Predict $Y$: “likes Pokémon”

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0.23</td>
<td>0.77</td>
</tr>
<tr>
<td>1</td>
<td>0.24</td>
<td>0.76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$X_2$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>5/13 ≈ 0.38</td>
<td>8/13 ≈ 0.62</td>
</tr>
<tr>
<td>1</td>
<td>7/17 ≈ 0.41</td>
<td>10/17 ≈ 0.59</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$Y$</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>13/30 ≈ 0.43</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>17/30 ≈ 0.57</td>
<td></td>
</tr>
</tbody>
</table>

Training data counts:

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$X_2$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$Y$</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>17</td>
<td></td>
</tr>
</tbody>
</table>
Training: Naïve Bayes for TV shows (MLE)

Observe indicator vars. $X = (X_1, X_2)$:
- $X_1$: “likes Star Wars”
- $X_2$: “likes Harry Potter”

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th></th>
<th>0</th>
<th>1</th>
<th></th>
<th>0</th>
<th>1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td></td>
<td></td>
<td></td>
<td>$X_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Y$</td>
<td>0</td>
<td>0.23</td>
<td>0.77</td>
<td>0</td>
<td>0.38</td>
<td>0.62</td>
<td>0</td>
<td>0.43</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.24</td>
<td>0.76</td>
<td>1</td>
<td>0.41</td>
<td>0.59</td>
<td>1</td>
<td>0.57</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Now that we’ve trained and found parameters, It’s time to classify new users!
Ex 1. Naïve Bayes Classifier (MLE)

\[
\hat{Y} = \arg\max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)
\]

Training

\forall i: \hat{P}(X_j = 1|Y = 0), \hat{P}(X_j = 0|Y = 0), \quad \text{Use MLE or Laplace (MAP)}
\hat{P}(X_j = 1|Y = 1), \hat{P}(X_j = 0|Y = 0),
\hat{P}(Y = 1), \hat{P}(Y = 0)

Testing

\[
\hat{Y} = \arg\max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)
\]
Testing: Naïve Bayes for TV shows (MLE)

Observe indicator vars. \( \mathbf{X} = (X_1, X_2) \):

<table>
<thead>
<tr>
<th>( X_1 )</th>
<th>( X_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y )</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.23 0.77</td>
</tr>
<tr>
<td>1</td>
<td>0.24 0.76</td>
</tr>
</tbody>
</table>

Predict \( Y \): “likes Pokémon”

Suppose a **new person** “likes Star Wars” (\( X_1 = 1 \)) but “dislikes Harry Potter” (\( X_2 = 0 \)).

Will they like Pokemon? Need to predict \( Y \):

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \hat{P}(\mathbf{X}|Y)\hat{P}(Y) = \arg \max_{y=\{0,1\}} \hat{P}(X_1|Y)\hat{P}(X_2|Y)\hat{P}(Y)
\]

If \( Y = 0 \):

\[
\hat{P}(X_1 = 1|Y = 0)\hat{P}(X_2 = 0|Y = 0)\hat{P}(Y = 0) = 0.77 \cdot 0.38 \cdot 0.43 = 0.126
\]

If \( Y = 1 \):

\[
\hat{P}(X_1 = 1|Y = 1)\hat{P}(X_2 = 0|Y = 1)\hat{P}(Y = 1) = 0.76 \cdot 0.41 \cdot 0.57 = 0.178
\]

Since term is greatest when \( Y = 1 \), predict \( \hat{Y} = 1 \)
Ex 2. Naïve Bayes Classifier (MAP)

\[
\hat{Y} = \arg \max_{y \in \{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j | Y) \right) \hat{P}(Y)
\]

Training:
\[
\forall i: \quad \hat{P}(X_j = 1 | Y = 0), \hat{P}(X_j = 0 | Y = 0), \quad \hat{P}(X_j = 1 | Y = 1), \hat{P}(X_j = 0 | Y = 0), \\
\hat{P}(Y = 1), \hat{P}(Y = 0)
\]

Use MLE or Laplace (MAP)

Testing:
\[
\hat{Y} = \arg \max_{y \in \{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j | Y) \right) \hat{P}(Y)
\]

(note the same as before)
Training: Naïve Bayes for TV shows (MAP)

Observe indicator vars. $X = (X_1, X_2)$:
- $X_1$: “likes Star Wars”
- $X_2$: “likes Harry Potter”

Predict $Y$: “likes Pokémon”

What are our MAP estimates using Laplace smoothing for $\hat{P}(X_j | Y)$?

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>0</th>
<th>1</th>
<th>$X_2$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y$</td>
<td></td>
<td></td>
<td>$Y$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>10</td>
<td>0</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>13</td>
<td>1</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

Training data counts

$\hat{P}(X_j = x | Y = y)$:

A. $\frac{\#(X_j=x,Y=y)}{\#(Y=y)}$

B. $\frac{\#(X_j=x,Y=y)+1}{\#(Y=y)+2}$

C. $\frac{\#(X_j=x,Y=y)+1}{\#(Y=y)+4}$

D. other
Training: Naïve Bayes for TV shows (MAP)

Observe indicator vars. $X = (X_1, X_2)$:
- $X_1$: “likes Star Wars”
- $X_2$: “likes Harry Potter”

Predict $Y$: “likes Pokémon”

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$Y$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>4</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$X_2$</th>
<th>$Y$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

Training data counts:

- $\hat{Y} = \arg \max_{Y=\{0,1\}} \left( \prod_{i=1}^{m} \hat{P}(X_i|Y) \right) \hat{P}(Y)$

In practice:
- We use Laplace for $\hat{P}(X_j|Y)$ in case some events $X_j = x_j$ don’t appear
- We **don’t** use Laplace for $\hat{P}(Y)$, because all class labels should appear reasonably often