23: Naïve Bayes

Jerry Cain
March 4th, 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

Deep Learning

Linear Regression

Naïve Bayes

Logistic Regression

Unbiased estimators

Maximizing likelihood

Bayesian estimation

\( \bar{X}, S^2 \)

\( \theta_{MLE} \)

\( \theta_{MAP} \)
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

Training Data

Prediction Function $\hat{\theta}$

Testing Data

Evaluation score

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Winter 2024

Stanford University
Supervised learning

Modeling

Real World Problem

Model the problem

Formal Model $\theta$

Learning Algorithm

Training Data

Testing Data

Prediction Function $\hat{\theta}$

Evaluation score

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

Prediction Function $\hat{\theta}$

Training Data

Evaluation score

Testing Data

Learning Algorithm

Training

Supervised learning is a basis for learning from data.
Model and dataset

Many different forms of machine learning
- We focus on a specific type of problem: prediction from 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
\[ \hat{Y} = g(X), \text{ a function on } X \]
Training data

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

<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>...</td>
</tr>
<tr>
<td>Patient 2</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Patient n</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
</tbody>
</table>
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

Real World Problem

Model the problem

Formal Model $\theta$

Learning Algorithm

Training Data

Testing Data

Prediction Function $\hat{\theta}$

Evaluation score
Testing data notation

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

\(n_{\text{test}}\) 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
Regression: Predicting real numbers

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

<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>...</td>
<td>1</td>
</tr>
<tr>
<td>Year 2</td>
<td>340.0</td>
<td>1</td>
<td>...</td>
<td>0</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>...</td>
<td>1</td>
</tr>
</tbody>
</table>

Global Land-Ocean temperature
Classification: Predicting class labels

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

<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>...</td>
</tr>
<tr>
<td>Patient 2</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patient n</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
</tbody>
</table>

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Winter 2024
Brute Force Bayes
Classification: Healthy hearts

$X = (X_1)$

<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>0</td>
</tr>
<tr>
<td>Patient n</td>
<td>1</td>
</tr>
</tbody>
</table>

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 \( X \)

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

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

(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

\[ X = (X_1) \]

| Feature 1 | Output | Conditional probability tables \( \hat{P}(X|Y) \) | Marginal probability table \( \hat{P}(Y) \) |
|-----------|--------|---------------------------------|---------------------------------|
| Patient 1 | 1      | \( \hat{P}(X|Y = 0) \) \( \theta_1 \) \( \theta_3 \) | \( \hat{P}(Y = 0) \) \( \theta_5 \) |
| Patient 2 | 1      | \( \hat{P}(X|Y = 0) \) \( \theta_2 \) \( \theta_4 \) | \( \hat{P}(Y = 1) \) \( \theta_6 \) |
| Patient n | 0      | \( \hat{P}(X|Y = 0) \) \( \theta_1 \) \( \theta_3 \) | \( \hat{P}(Y = 0) \) \( \theta_5 \) |

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

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

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

Count: # datapoints

$\hat{P}(X|Y = 0)$  $\hat{P}(X|Y = 1)$

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$\theta_1$</th>
<th>$\theta_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1 = 0$</td>
<td>$\theta_1$</td>
<td>$\theta_3$</td>
</tr>
<tr>
<td>$X_1 = 1$</td>
<td>$\theta_2$</td>
<td>$\theta_4$</td>
</tr>
</tbody>
</table>

$X|Y = 0$ and $X|Y = 1$

are each multinomials with 2 outcomes!

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                 | 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!

<table>
<thead>
<tr>
<th>Count:</th>
<th># datatpoints</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, \( \hat{P}(X|Y) \)

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

| \( \hat{P}(X|Y = 0) \) | \( \hat{P}(X|Y = 1) \) |
|----------------------|----------------------|
| \( X_1 = 0 \) | 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!

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

Just count + add imaginary trials!
### 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 $\hat{P}(X_1 = x|Y = y)$**

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

**Just count!**

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

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

**Just count + add imaginary trials!**

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

**Count:**

- $X_1 = 0, Y = 0$: 4
- $X_1 = 1, Y = 0$: 6
- $X_1 = 0, Y = 1$: 0
- $X_1 = 1, Y = 1$: 100
- Total: 110
Testing

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

| (MAP) | \(\hat{P}(X|Y = 0)\) | \(\hat{P}(X|Y = 1)\) | (MLE) | \(\hat{P}(Y)\) |
|-------|-----------------|-----------------|-------|-----------------|
| \(X_1 = 0\) | 0.42            | 0.01            | \(Y = 0\) | 0.09            |
| \(X_1 = 1\) | 0.58            | 0.99            | \(Y = 1\) | 0.91            |

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

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

A. \(0.052 < 0.5 \quad \Rightarrow \quad \hat{Y} = 1\)
B. \(0.901 > 0.5 \quad \Rightarrow \quad \hat{Y} = 1\)
C. \(0.052 < 0.901 \quad \Rightarrow \quad \hat{Y} = 1\)
Brute Force Bayes classifier

\[ \hat{Y} = \text{arg max} \left( \hat{P}(X|Y) \hat{P}(Y) \right) \quad y = \{0,1\} \]

(\(\hat{P}(Y)\) 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)

Training

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

\[ \hat{Y} = \text{arg max} \left( \hat{P}(X_1, X_2, \ldots, X_m|Y) \hat{P}(Y) \right) \quad y = \{0,1\} \]

Testing

\(\hat{P}(X_1, X_2, \ldots, X_m|Y = 1)\)
\(\hat{P}(X_1, X_2, \ldots, X_m|Y = 0)\)
\(\hat{P}(Y = 1)\)
\(\hat{P}(Y = 0)\)
Naïve Bayes
Brute Force Bayes: $m = 300$ (# features)

$$X = (X_1, X_2, X_3, ..., X_{300})$$

<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>...</td>
</tr>
<tr>
<td>Patient 2</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Patient n</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
</tbody>
</table>

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

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

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

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

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

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

- \( \hat{P}(Y = 1 | x) \): 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. \( 2 \cdot 2 + 2 = 6 \)
B. \( 2 \cdot 300 + 2 = 602 \)
C. \( 2 \cdot 2^{300} + 2 = \text{a lot} \)
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

- $\hat{P}(Y = 1 | x)$: 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. $2 \cdot 2 + 2 = 6$
B. $2 \cdot 300 + 2 = 602$
C. $2 \cdot 2^{300} + 2 = \text{a lot}$

This approach requires you to learn $O(2^m)$ parameters.
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/\( \hat{P}(X) \) 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:

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

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

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**

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

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

\( \hat{P}(Y = 1) \)

**Testing**

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

Use MLE or Laplace (MAP)
NETFLIX

and Learn
### Classification terminology check

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

#### A. \(x^{(i)}\)

#### B. \(y^{(i)}\)

#### C. \((x^{(i)}, y^{(i)})\)

#### D. \(x_j\)

1: like movie  
0: dislike movie

<table>
<thead>
<tr>
<th></th>
<th>Movie 1</th>
<th>Movie 2</th>
<th>Movie (m)</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>User 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>User (n)</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
</tr>
</tbody>
</table>

User 1: 1: like movie, 0: dislike movie

User 2: 1: like movie, 0: dislike movie

User \(n\): 1: like movie, 0: dislike movie
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 $\hat{Y} = \underset{y=\{0,1\}}{\text{arg max}} \hat{P}(Y | X)$

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Winter 2024
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

\[
\hat{P}(X_1 = 1, X_2 = 1|Y = 0) \text{ with MLE? with Laplace?}
\]

• Brute Force Bayes
• Naïve Bayes

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

Naïve Bayes Assumption

\[
P(X|Y) = \prod_{j=1}^{m} P(X_j|Y)
\]
**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**

\( \forall i: \hat{P}(X_j = 1|Y = 0), \hat{P}(X_j = 0|Y = 0), \) 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. \( X = (X_1, X_2) \):
- \( X_1 \): “likes Star Wars”
- \( X_2 \): “likes Harry Potter”

Predict \( Y \): “likes Pokémon”

<table>
<thead>
<tr>
<th>( X_1 )</th>
<th>( Y ) = 0</th>
<th>( Y ) = 1</th>
</tr>
</thead>
<tbody>
<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>( Y ) = 0</th>
<th>( Y ) = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

Training data counts

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

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

| \( Y \) | \( \hat{P}(X_1 = 0 | Y) \) | \( \hat{P}(X_1 = 1 | Y) \) |
|---|---|---|
| 0 | \( \hat{P}(X_1 = 0 | Y = 0) \) | \( \hat{P}(X_1 = 1 | Y = 0) \) |
| 1 | \( \hat{P}(X_1 = 0 | Y = 1) \) | \( \hat{P}(X_1 = 1 | Y = 1) \) |
**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>$Y$</th>
<th>$0$</th>
<th>$1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0$</td>
<td></td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>$1$</td>
<td></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>$0$</td>
<td></td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>$1$</td>
<td></td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

Training data counts

1. How many datapoints ($n$) are in our training data?
2. Compute MLE estimates for $\hat{P}(X_1 | Y)$:
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></th>
<th>X_1</th>
<th></th>
<th>X_2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>10</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>13</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

Training data counts

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

(from last slide)

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Winter 2024
### Training: Naïve Bayes for TV shows (MLE)

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

<table>
<thead>
<tr>
<th>( X_1 )</th>
<th>( Y )</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>0.23 0.77</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0.24 0.76</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></td>
<td>0</td>
<td>0.38 0.62</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0.41 0.59</td>
</tr>
</tbody>
</table>

Predict \( Y \): “likes Pokémon”

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**

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

**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. \( X = (X_1, X_2) \):

- \( X_1 \): “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>0</td>
<td>0.23 0.77</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.24 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>0</td>
<td>0.38 0.62</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.41 0.59</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( Y )</th>
<th>0</th>
<th>0.43</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>0.57</td>
</tr>
</tbody>
</table>

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

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

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

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)

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

Training

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

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

(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”

<table>
<thead>
<tr>
<th>( X_1 )</th>
<th>( Y )</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X_1' )</td>
<td>( Y )</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>( X_1'' )</td>
<td>( Y )</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>( X_2' )</td>
<td>( Y )</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>( X_2'' )</td>
<td>( Y )</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

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

\[
\hat{P}(X_j = x \mid Y = y) = \frac{\#(X_j = x, Y = y)}{\#(Y = y)}
\]

- **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”

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

Training data counts

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

\[
\begin{array}{c|cc}
X_2 & 0 & 1 \\
\hline
Y & 0 & 0.27 & 0.73 \\
0 & 0.40 & 0.60 \\
1 & 0.42 & 0.58 \\
\end{array}
\]

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
and Learn naively
What is Bayes doing in my mail server?

Let’s get Bayesian on your spam:

Content analysis details: (49.5 hits, 7.0 required)
0.9 RCVD_IN_PBL RBL: Received via a relay in Spamhaus PBL [93.40.189.29 listed in zen.spamhaus.org]
1.5 URIBL_WS_SURBL Contains an URL listed in the WS SURBL blocklist [URIs: recragas.cn]
5.0 URIBL_JP_SURBL Contains an URL listed in the JP SURBL blocklist [URIs: recragas.cn]
5.0 URIBL_OB_SURBL Contains an URL listed in the OB SURBL blocklist [URIs: recragas.cn]
5.0 URIBL_SC_SURBL Contains an URL listed in the SC SURBL blocklist [URIs: recragas.cn]
2.0 URIBL_BLACK Contains an URL listed in the URIBL blacklist [URIs: recragas.cn]
8.0 BAYES_99 BODY: Bayesian spam probability is 99 to 100% [score: 1.0000]
Email classification

Goal
Based on email content $X$, predict if email is spam or not.

Features
Consider a lexicon $m$ words (for English: $m \approx 100,000$).
$X = (X_1, X_2, ..., X_m)$, $m$ indicator variables
$X_j = 1$ if word $j$ appeared in document

Output
$Y = 1$ if email is spam

Note: $m$ is huge. Make Naïve Bayes assumption: $P(X|\text{spam}) = \prod_{j=1}^{m} P(X_j|\text{spam})$

Appearances of words in email are conditionally independent given the email is spam or not
Training: Naïve Bayes Email classification

Train set

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

\[ x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_m^{(i)}) \text{ for each word, whether it appears in email } i \]

\[ y^{(i)} = 1 \text{ if spam, 0 if not spam} \]

Note: \( m \) is huge.

Which estimator should we use for \( \hat{P}(X_j | Y) \)?

A. MLE
B. Laplace estimate (MAP)
C. Other MAP estimate
D. Both A and B

Many words are likely to not appear at all in the training set!
Ex 3. Naïve Bayes Classifier ($m, n$ large)

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

Training

forall $j$: $\hat{P}(X_j = 1 | Y = 0), \hat{P}(X_j = 0 | Y = 0)$,
$\hat{P}(X_j = 1 | Y = 1), \hat{P}(X_j = 0 | Y = 0)$,
$\hat{P}(Y = 1), \hat{P}(Y = 0)$

Use MLE or Laplace (MAP)

Testing

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

Laplace (MAP) estimates avoid estimating 0 probabilities for events that don’t occur in your training data.
Testing: Naïve Bayes Email classification

For a new email:
- Generate $X = (X_1, X_2, ..., X_m)$
- Classify as spam or not using Naïve Bayes assumption

Note: $m$ is huge.

Suppose train set size $n$ also huge (many labeled emails).
Can we still use the below prediction?

$$
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)
$$
Testing: Naïve Bayes Email classification

For a new email:
  • Generate $X = (X_1, X_2, ..., X_m)$
  • Classify as spam or not using Naïve Bayes assumption

Note: $m$ is huge.

Suppose train set size $n$ also huge (many labeled emails).
Can we still use the below prediction?

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

Will probably lead to underflow!
Ex 3. Naïve Bayes Classifier \((m, n \text{ large})\)

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

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

Use sums of log-probabilities for numerical stability.

Testing

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \log \hat{P}(Y) + \sum_{j=1}^{m} \log \hat{P}(X_j|Y) \right)
\]
How well does Naïve Bayes perform?

After training, you can test with another set of data, called the test set.
- Test set also has known values for $Y$ so we can see how often we were right/wrong in our predictions $\hat{Y}$.

Typical workflow:
- Have a dataset of 1789 emails (1578 spam, 211 ham)
- Train set: First 1538 emails (by time)
- Test set: Next 251 messages

Evaluation criteria on test set:

<table>
<thead>
<tr>
<th></th>
<th>Spam Prec.</th>
<th>Spam Recall</th>
<th>Non-spam Prec.</th>
<th>Non-spam Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words only</td>
<td>97.1%</td>
<td>94.3%</td>
<td>87.7%</td>
<td>93.4%</td>
</tr>
<tr>
<td>Words + addtl features</td>
<td>100%</td>
<td>98.3%</td>
<td>96.2%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Winter 2024
What are precision and recall?

Accuracy (# correct)/(# total) sometimes just doesn’t cut it.

**Precision**: Of the emails you predicted as spam, how many are *truly* spam?  
Measure of false positives

**Recall**: Of the emails that are truly spam, how many did you predict?  
Measure of false negatives