Naïve Bayes
Chris Piech
CS109, Stanford University
Review
MLE vs MAP

Maximum Likelihood Estimation

\[ \theta_{\text{MLE}} = \arg \max_{\theta} f(X_1, X_2, \ldots, X_n | \theta) \]
\[ = \arg \max_{\theta} \sum_i \log f(X_i | \theta) \]

Maximum A Posteriori

\[ \theta_{\text{MAP}} = \arg \max_{\theta} f(\theta | X_1, X_2, \ldots, X_n) \]
\[ = \arg \max_{\theta} \left( \log g(\theta) + \sum_i \log f(X_i | \theta) \right) \]
Multinomial with Laplace Smoothing

\[ \theta_{\text{MAP}} = \underset{\theta}{\text{argmax}} \ f(\theta|X_1, X_2, \ldots, X_n) \]

\[ = \underset{\theta}{\text{argmax}} \ (\log g(\theta) + \sum_{i} \log f(X_i|\theta)) \]

\[ p_i = \frac{x_i + 1}{n + m} \]

MAP estimate of the probability of outcome \( i \)

Number of observed outcomes of type \( i \)

Number of observations

Number of outcome types
The last estimator has risen...
Machine Learning
Many different forms of “Machine Learning”

- We focus on the problem of prediction

Want to make a prediction based on observations

- Vector $X$ of $m$ observed variables: $X = [X_1 \ldots X_m]$
- Based on observed $X$, want to predict unseen variable $Y$
  - $Y$ called “output feature/variable” (or the “dependent variable”)
- Seek to “learn” a function $g(X)$ to predict $Y$:
  - $\hat{Y} = g(X)$
  - When $Y$ is discrete, prediction of $Y$ is called “classification”
  - When $Y$ is continuous, prediction of $Y$ is called “regression”
Training Data

Assume IID data:

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

\[m = |x^{(i)}|\]

Each datapoint has m features and a single output
Example Datasets

Heart

Ancestry

Netflix
<table>
<thead>
<tr>
<th>Movie 1</th>
<th>Movie 2</th>
<th>Movie ( m )</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Independence Day" /></td>
<td><img src="image2" alt="The Patriot" /></td>
<td><img src="image3" alt="Other Movies" /></td>
<td><img src="image4" alt="Output Movies" /></td>
</tr>
<tr>
<td>User 1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>User 2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>User ( n )</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
**Single Instance**

<table>
<thead>
<tr>
<th>Movie 1</th>
<th>Movie 2</th>
<th>Movie $m$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Poster" /></td>
<td><img src="image2" alt="Poster" /></td>
<td><img src="image3" alt="Poster" /></td>
<td><img src="image4" alt="Poster" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User 1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>User $n$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$(x^{(i)}, y^{(i)})$ such that $1 \leq i \leq n$
A feature vector can be represented as a matrix with movies as columns and users as rows. Each element $(x^{(i)}, y^{(i)})$ in the matrix represents a user's rating for a particular movie, such that $1 \leq i \leq n$.

<table>
<thead>
<tr>
<th>User 1</th>
<th>Movie 1</th>
<th>Movie 2</th>
<th>Movie $m$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User 2</th>
<th>Movie 1</th>
<th>Movie 2</th>
<th>Movie $m$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User $n$</th>
<th>Movie 1</th>
<th>Movie 2</th>
<th>Movie $m$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>User 1</td>
<td>Movie 1</td>
<td>Movie 2</td>
<td>Movie $m$</td>
<td>Output</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>---------</td>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>User 2</td>
<td>1</td>
<td>1</td>
<td>0</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>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$(x^{(i)}, y^{(i)})$ such that $1 \leq i \leq n$
### Single Feature Value

<table>
<thead>
<tr>
<th>Movie 1</th>
<th>Movie 2</th>
<th>Movie $m$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="movie1.png" alt="Image" /></td>
<td><img src="movie2.png" alt="Image" /></td>
<td><img src="movie3.png" alt="Image" /></td>
<td><img src="output.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User 1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>User $n$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

In general: $x_j^i$

In this case: $x_m^2$
# Healthy Heart Classifier

<table>
<thead>
<tr>
<th></th>
<th>ROI 1</th>
<th>ROI 2</th>
<th>ROI ( m )</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heart 1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Heart 2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Heart ( n )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

![Heart Images]
## Healthy Heart Classifier

<table>
<thead>
<tr>
<th>Heart 1</th>
<th>ROI 1</th>
<th>ROI 2</th>
<th>ROI $m$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heart 1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Heart 2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Heart $n$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

$x_2^{(1)}$
## Healthy Heart Classifier

<table>
<thead>
<tr>
<th>Heart 1</th>
<th>ROI 1</th>
<th>ROI 2</th>
<th>ROI ( m )</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heart 1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Heart 2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Heart ( n )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ (x^{(2)}, y^{(2)}) \]
Healthy Heart Classifier

<table>
<thead>
<tr>
<th>ROI 1</th>
<th>ROI 2</th>
<th>ROI ( m )</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="ROI 1" /></td>
<td><img src="image2.png" alt="ROI 2" /></td>
<td><img src="image3.png" alt="ROI ( m )" /></td>
<td><img src="image4.png" alt="Output" /></td>
</tr>
</tbody>
</table>

| Heart 1 | 0 | 1 | 1 | 0 |
| Heart 2 | 1 | 1 | 1 | 0 |
| Heart \( n \) | 0 | 0 | 0 | 1 |

\[ x^{(2)} \]

\( x^{(2)} \):
- Heart 1: 0, 1, 1, 0
- Heart 2: 1, 1, 1, 0
- Heart \( n \): 0, 0, 0, 1
Healthy Heart Classifier

<table>
<thead>
<tr>
<th>ROI 1</th>
<th>ROI 2</th>
<th>ROI ( m )</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.jpg" alt="ROI 1" /></td>
<td><img src="image2.jpg" alt="ROI 2" /></td>
<td><img src="image3.jpg" alt="ROI ( m )" /></td>
<td>( 0 )</td>
</tr>
<tr>
<td>Heart 1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Heart 2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Heart ( n )</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\( y^{(2)} \)
Ancestry Classifier

SNP 1   SNP 2   SNP \( m \)   Output

| User 1 | 1 | 0 | 1 | 0 |
| User 2 | 0 | 0 | 1 | 1 |
| User \( n \) | 1 | 1 | 0 | 1 |
# How Many Points will Warriors Score

<table>
<thead>
<tr>
<th>Opposing team</th>
<th>Points in last game</th>
<th>At Home?</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game 1 84</td>
<td>105</td>
<td>1</td>
<td>120</td>
</tr>
<tr>
<td>Game 2 90</td>
<td>102</td>
<td>0</td>
<td>95</td>
</tr>
<tr>
<td>Game n 74</td>
<td>120</td>
<td>0</td>
<td>115</td>
</tr>
</tbody>
</table>

Opposing team ELO Points in last game At Home? Output

GAME 1

GAME 2
Training Data

Assume IID data:

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

\[m = |x^{(i)}|\]

Each datapoint has m features and a single output
ML is ubiquitous
Linear Regression
A Grounding Example: Linear Regression

Problem: Predict real value \( Y \) based on observing variable \( X \)

\( N \) training pairs: \((x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})\)

Model: Linear weight for every feature

\[
\hat{Y} = \theta_1 X_1 + \theta_2 X_2 + \ldots \theta_{n-1} X_{n-1} + \theta_n 1 \\
= \theta^T X
\]

Training: Chose the best thetas to describe your data

\[
\hat{\theta}_{MLE} = \arg\max_{\theta} \sum_{i=1}^{n} (y^{(i)} - \theta^T x^{(i)})^2
\]

Use Gradient Ascent to optimize

MLE based on assumption

\[
Y = \hat{Y} + Z \\
Z \sim N(0, \sigma^2)
\]
Predicting Warriors

\[ X_1 = \text{Opposing team ELO} \]
\[ X_2 = \text{Points in last game} \]
\[ X_3 = \text{Curry playing?} \]
\[ X_4 = \text{Playing at home?} \]

\[ Y = \text{Warriors points} \]
Predicting CO$_2$

$X_1 =$ Temperature

$X_2 =$ Elevation

$X_3 =$ CO$_2$ level yesterday

$X_4 =$ GDP of region

$X_5 =$ Acres of forest growth

$Y =$ CO$_2$ levels
**Training data:** set of $n$ pre-classified data instances

- $n$ training pairs: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})$
  - Use superscripts to denote $i$-th training instance

**Learning algorithm:** method for determining $g(X)$

- Given a new input observation of $x = x_1, x_2, \ldots, x_m$
  - Use $g(x)$ to compute a corresponding output (prediction)
Predicting Warriors

$Y = \text{Warriors points}$

$$\hat{Y} = \theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_{n-1} X_{n-1} + \theta_n 1$$

$$= \theta^T X .$$

$X_1 = \text{Opposing team ELO}$

$X_2 = \text{Points in last game}$

$X_3 = \text{Curry playing?}$

$X_4 = \text{Playing at home?}$

$\theta_1 = -2.3$

$\theta_2 = +1.2$

$\theta_3 = +10.2$

$\theta_4 = +3.3$

$\theta_5 = +95.4$
Classification
# Healthy Heart Classifier

<table>
<thead>
<tr>
<th>ROI 1</th>
<th>ROI 2</th>
<th>ROI (m)</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Heart 1 ROI 1" /></td>
<td><img src="image2" alt="Heart 1 ROI 2" /></td>
<td><img src="image3" alt="Heart 1 ROI m" /></td>
<td>0</td>
</tr>
<tr>
<td><img src="image4" alt="Heart 2 ROI 1" /></td>
<td><img src="image5" alt="Heart 2 ROI 2" /></td>
<td><img src="image6" alt="Heart 2 ROI m" /></td>
<td>0</td>
</tr>
<tr>
<td><img src="image7" alt="Heart n ROI 1" /></td>
<td><img src="image8" alt="Heart n ROI 2" /></td>
<td><img src="image9" alt="Heart n ROI m" /></td>
<td>1</td>
</tr>
</tbody>
</table>

The table above shows the input ROI images for different hearts and their corresponding outputs.
## Ancestry Classifier

<table>
<thead>
<tr>
<th>SNP 1</th>
<th>SNP 2</th>
<th>SNP ( m )</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="genotype1.png" alt="Genotype" /></td>
<td><img src="genotype2.png" alt="Genotype" /></td>
<td><img src="genotype3.png" alt="Genotype" /></td>
<td><img src="output.png" alt="Output Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User 1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>User ( n )</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
NETFLIX
And Learn
## Target Movie “Like” Classification

<table>
<thead>
<tr>
<th>Feature 1</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>1 *</td>
</tr>
<tr>
<td>User 2</td>
<td>1 *</td>
</tr>
<tr>
<td>User (n)</td>
<td>0 *</td>
</tr>
</tbody>
</table>

\[
x_j^{(i)} \in \{0, 1\} \quad \text{and} \quad y^{(i)} \in \{0, 1\}
\]
How could we predict the class label: will the user like life is beautiful?
Fake Algorithm: Brute Bayes Classifier
Brute Force Bayes

Prediction: will they like L.I.B.?

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

Whether or not they liked Independence Day

If \( y = 1 \), they like L.I.B.?

Simply chose the class label that is the most likely given the data

This is for one user
Brute Force Bayes

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

Simply chose the class label that is the most likely given the data.

This is for one user.
Brute Force Bayes

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

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

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

Simply chose the class label that is the most likely given the data

This is for one user

* Note how similar this is to Hamilton example 😊
What are the Parameters?
Brute Force Bayes

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

Learn these during training

Joint probability table
Training

Let \((x_1, y)\) be a multinomial with four outcomes.
<table>
<thead>
<tr>
<th>User</th>
<th>$x_1$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>User 2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>User $n$</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.2</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>0.3</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**MLE:** Just count
### MAP Estimate

<table>
<thead>
<tr>
<th>User</th>
<th>$x_1$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>User 2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>User $n$</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$x_1 = 0$</th>
<th>$x_1 = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y = 0$</td>
<td>0.20</td>
<td>0.01</td>
</tr>
<tr>
<td>$y = 1$</td>
<td>0.30</td>
<td>0.49</td>
</tr>
</tbody>
</table>

Add Laplace smoothing
$$\hat{y} = \arg\max_{y=\{0,1\}} P(x|y)P(y)$$

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.20</td>
<td>0.01</td>
<td></td>
<td>0.21</td>
</tr>
<tr>
<td>1</td>
<td>0.30</td>
<td>0.49</td>
<td></td>
<td>0.79</td>
</tr>
</tbody>
</table>

Test user: Likes independence day

$$P(x_1 = 1|y = 0)P(y = 0)$$

vs

$$P(x_1 = 1|y = 1)P(y = 1)$$
That was pretty good!
**Brute Force Bayes** \( m = 2 \)

<table>
<thead>
<tr>
<th>User</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>User 2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>User ( n )</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Simply chose the class label that is the most likely given the data

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

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

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

\[
P(x_1, x_2 | y)
\]
Brute Force Bayes

\[ \hat{y} = \arg \max_y P(x|y) P(y) \]
\[ y = \{0, 1\} \]

\[ \begin{array}{c|cc}
X_1 & 0 & 1 \\
\hline
0 & \theta_0 & \theta_1 \\
1 & \theta_2 & \theta_3 \\
\end{array} \]

\[ \begin{array}{c|cc}
X_1 & 0 & 1 \\
\hline
0 & \theta_4 & \theta_5 \\
1 & \theta_6 & \theta_7 \\
\end{array} \]
Fine
### Brute Force Bayes $m = 3$

<table>
<thead>
<tr>
<th>User</th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>User 2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>User $n$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Simply chose the class label that is the most likely given the data

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

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

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

\[
P(x_1, x_2, x_3|y)
\]
\[ \hat{y} = \arg\max_y P(x|y)P(y) \]
\[ y = \{0, 1\} \]

<table>
<thead>
<tr>
<th></th>
<th>$X_2 = 0$</th>
<th>$X_2 = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$X_1$</td>
<td>$X_1$</td>
</tr>
<tr>
<td>$Y$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>$\theta_0$</td>
<td>$\theta_1$</td>
</tr>
<tr>
<td>1</td>
<td>$\theta_2$</td>
<td>$\theta_3$</td>
</tr>
<tr>
<td>$X_3 = 0$</td>
<td>$X_2 = 0$</td>
<td>$X_2 = 1$</td>
</tr>
<tr>
<td></td>
<td>$X_1$</td>
<td>$X_1$</td>
</tr>
<tr>
<td>$Y$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>$\theta_4$</td>
<td>$\theta_5$</td>
</tr>
<tr>
<td>1</td>
<td>$\theta_6$</td>
<td>$\theta_7$</td>
</tr>
</tbody>
</table>
And if \( m = 100 \)?
Simply chose the class label that is the most likely given the data

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

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

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

\[ P(x_1, x_2, x_3, \ldots, x_{100}|y) \]
Oops... Number of atoms in the universe
What is the big O for # parameters?

\[ m = \# \text{ features}. \]
What is the big O for # parameters?

\( m = \# \text{ features.} \)

\[ O(2^m) \]

Assuming each feature is binary...
Not going to cut it!
\[ \hat{y} = \arg\max_{y=\{0,1\}} P(y|x) \]

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

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

---

\[ P(x|y) = P(x_1, x_2, \ldots, x_m|y) \]
Naïve Bayes Assumption

\[ \hat{y} = \text{argmax}_{y \in \{0, 1\}} P(y|x) \]

\[ = \text{argmax}_{y \in \{0, 1\}} \frac{P(x|y)P(y)}{P(x)} \]

\[ = \text{argmax}_{y \in \{0, 1\}} P(x|y)P(y) \]

\[ P(x|y) = P(x_1, x_2, \ldots, x_m|y) \]

\[ = \prod_{i} P(x_i|y) \]

The Naïve Bayes assumption
Naïve Bayes Assumption:

\[ P(x|y) = \prod_i P(x_i|y) \]
Naïve Bayes Classifier
Naïve Bayes Classifier

- Say, we have $m$ input values $\mathbf{X} = <X_1, X_2, \ldots, X_m>$
  - Assume variables $X_1, X_2, \ldots, X_m$ are **conditionally independent** given $Y$
    - Really don’t believe $X_1, X_2, \ldots, X_m$ are conditionally independent
    - Just an approximation we make to be able to make predictions
    - This is called the “Naive Bayes” assumption, hence the name
  - Predict $Y$ using $\hat{Y} = \arg \max_y P(\mathbf{X}, Y) = \arg \max_y P(\mathbf{X} | Y)P(Y)$
    - But, we now have:
      $$P(\mathbf{X} | Y) = P(X_1, X_2, \ldots X_m | Y) = \prod_{i=1}^{m} P(X_i | Y)$$ by conditional independence
  - Note: computation of PMF table is **linear** in $m$ : $O(m)$
    - Don’t need much data to get good probability estimates
Predict \( Y \) based on observing variables \( X_1 \) and \( X_2 \)
- \( X_1 \) and \( X_2 \) are both indicator variables
  - \( X_1 \) denotes “likes Star Wars”, \( X_2 \) denotes “likes Harry Potter”
- \( Y \) is indicator variable: “likes Lord of the Rings”
  - Use training data to estimate PMFs: \( \hat{P}_{X_i,Y}(x_i, y), \hat{P}_Y(y) \)

<table>
<thead>
<tr>
<th>( X_1 )</th>
<th>( Y )</th>
<th>0</th>
<th>1</th>
<th>MLE estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>3</td>
<td>10</td>
<td>0.10 0.33</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>4</td>
<td>13</td>
<td>0.13 0.43</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( X_2 )</th>
<th>( Y )</th>
<th>0</th>
<th>1</th>
<th>MLE estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>5</td>
<td>8</td>
<td>0.17 0.27</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>7</td>
<td>10</td>
<td>0.23 0.33</td>
</tr>
</tbody>
</table>

- Say someone likes Star Wars (\( X_1 = 1 \)), but not Harry Potter (\( X_2 = 0 \))
- Will they like “Lord of the Rings”? Need to predict \( Y \):

\[
\hat{y} = \arg \max_y \hat{P}(X \mid Y)\hat{P}(Y) = \arg \max_y \hat{P}(X_1 \mid Y)\hat{P}(X_2 \mid Y)\hat{P}(Y)
\]


One SciFi/Fantasy to Rule them All

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>0</th>
<th>1</th>
<th>MLE estimates</th>
<th>$X_2$</th>
<th>0</th>
<th>1</th>
<th>MLE estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>10</td>
<td>0.10 0.33</td>
<td>0</td>
<td>5</td>
<td>8</td>
<td>0.17 0.27</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>13</td>
<td>0.13 0.43</td>
<td>1</td>
<td>7</td>
<td>10</td>
<td>0.23 0.33</td>
</tr>
</tbody>
</table>

- **Prediction for $Y$ is value of $Y$ maximizing $P(X, Y)$:**
  \[
  \hat{Y} = \arg\max_y \hat{P}(X \mid Y) \hat{P}(Y) = \arg\max_y \hat{P}(X_1 \mid Y) \hat{P}(X_2 \mid Y) \hat{P}(Y)
  \]

  - **Compute $P(X, Y=0)$:**
    \[
    \hat{P}(X_1 = 1 \mid Y = 0) \hat{P}(X_2 = 0 \mid Y = 0) \hat{P}(Y = 0) \\
    \frac{\hat{P}(X_1 = 1, Y = 0)}{\hat{P}(Y = 0)} \frac{\hat{P}(X_2 = 0, Y = 0)}{\hat{P}(Y = 0)} \hat{P}(Y = 0) \\
    \approx \frac{0.33}{0.43} \frac{0.17}{0.43} 0.43 \approx 0.13
    \]
  
  - **Compute $P(X, Y=1)$:**
    \[
    \hat{P}(X_1 = 1 \mid Y = 1) \hat{P}(X_2 = 0 \mid Y = 1) \hat{P}(Y = 1) \\
    \frac{\hat{P}(X_1 = 1, Y = 1)}{\hat{P}(Y = 1)} \frac{\hat{P}(X_2 = 0, Y = 1)}{\hat{P}(Y = 1)} \hat{P}(Y = 1) \\
    \approx \frac{0.43}{0.57} \frac{0.23}{0.57} 0.57 \approx 0.17
    \]

- **Since $P(X, Y=1) > P(X, Y=0)$, we predict $\hat{Y} = 1$**
What is Bayes Doing in my Mail Server

• This is spam:

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]

A Bayesian Approach to Filtering Junk E-Mail

Mehran Sahami*$\quad$ Susan Dumais†$\quad$ David Heckerman†$\quad$ Eric Horvitz†

$^*\text{Gates Building 1A}$
Computing Science Department
Stanford University
Stanford, CA 94305-9010
sahami@cs.stanford.edu

$^\dagger\text{Microsoft Research}$
Redmond, WA 98052-6399
{sudumais, heckerman, horvitz}@microsoft.com

Abstract

In addressing the growing problem of junk E-mail on the Internet, we examine methods for the automated
count material (such as graphic pornography), there is often a higher cost to users of actually
viewing this mail than simply the time to sort out the

*first line of text is cut off
Spam, Spam... Go Away!

• The constant battle with spam

“And machine-learning algorithms developed to merge and rank large sets of Google search results allow us to combine hundreds of factors to classify spam.”

Source: http://www.google.com/mail/help/fightspm/spamexplained.html
Email Classification

- Want to predict if an email is spam or not
  - Start with the input data
    - Consider a lexicon of $m$ words (Note: in English $m \approx 100,000$)
    - Define $m$ indicator variables $X = <X_1, X_2, \ldots, X_m>$
    - Each variable $X_i$ denotes if word $i$ appeared in a document or not
      - Note: $m$ is huge, so make “Naive Bayes” assumption
  - Define output classes $Y$ to be: \{spam, non-spam\}
  - Given training set of $N$ previous emails
    - For each email message, we have a training instance: $X = <X_1, X_2, \ldots, X_m>$ noting for each word, if it appeared in email
    - Each email message is also marked as spam or not (value of $Y$)
Training the Classifier

- Given \( N \) training pairs: 
  \[(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})\]

- Learning
  - Estimate probabilities \( P(Y) \) and each \( P(X_i \mid Y) \) for all \( i \)
    - Many words are likely to not appear at all in given set of email
  - Laplace estimate: 
    \[ \hat{p}(X_i = 1 \mid Y = \text{spam})_{\text{Laplace}} = \frac{(#\text{spam emails with word } i) + 1}{\text{total # spam emails} + 2} \]

- Classification
  - For a new email, generate \( \mathbf{X} = <X_1, X_2, \ldots, X_m> \)
  - Classify as spam or not using:
    \[ \hat{Y} = \arg \max_y \hat{P}(\mathbf{X} \mid Y) \hat{P}(Y) \]
  - Employ Naive Bayes assumption: 
    \[ \hat{P}(\mathbf{X} \mid Y) = \prod_{i=1}^{m} \hat{P}(X_i \mid Y) \]
Training Naïve Bayes, is estimating parameters for a multinomial.

Thus it is just counting.
After training, can test with another set of data

- “Testing” set also has known values for Y, so we can see how often we were right/wrong in predictions for Y

- Spam data
  - Email data set: 1789 emails (1578 spam, 211 non-spam)
  - First, 1538 email messages (by time) used for training
  - Next 251 messages used to test learned classifier

- Criteria:
  - Precision $= \#\text{ correctly predicted class } Y / \#\text{ predicted class } Y$
  - Recall $= \#\text{ correctly predicted class } Y / \#\text{ real class } Y\text{ messages}$

<table>
<thead>
<tr>
<th></th>
<th>Spam</th>
<th>Non-spam</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>Words only</td>
<td>97.1%</td>
<td>94.3%</td>
</tr>
<tr>
<td>Words + add’l features</td>
<td>100%</td>
<td>98.3%</td>
</tr>
</tbody>
</table>
On biased datasets
Ethics and Datasets?

Sometimes machine learning feels universally unbiased.

We can even prove our estimators are “unbiased” 😊

Google/Nikon/HP had biased datasets
Ancestry dataset prediction

East Asian

or

Ad Mixed American (Native, European and African Americans)
It is much easier to write a binary classifier when learning ML for the first time.
Learn Three Things From This


2. That genetic ancestry paints a more realistic picture of how we are mixed in many nuanced ways.

3. The importance of choosing the right data to learn from. Your results will be as biased as your dataset.

Know it so you can beat it!
Ethics in Machine Learning is a whole new field