Classification – Naïve Bayes

Introduction

M. R. Hasan
Readings

• Bishop: 2.1, 2.2
• Murphy: 1.2, 1.3, 2.3.1, 2.3.2, 3.5, 3.5.1, 3.5.2, 3.5.3, 3.5.4, 3.5.5
What We Will Cover

• Supervised Learning: Classification
• Generative Learning Model
• Bayes’ Optimal Classifier & Its Complexity
• Naïve Bayes Assumption & Naïve Bayes Classifier
Supervised Learning

• In a **supervised** learning problem, for each sample data attribute (e.g., size of living area), we have its **label** (e.g., house price) as well.

• A pair \((x_i, y_i)\) is called a **training example**.

• The dataset that we will use to learn is a list of \(N\) **training examples**: \(\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}\)
Supervised Learning

• When the target variable that we are trying to predict is continuous, we call the learning problem a regression problem.

• When $y$ can take on only a small number of discrete values, we call it a classification problem.
Classification

- Consider a classification problem in which we want to learn to distinguish between **elephants** \((y = 1)\) and **dogs** \((y = 0)\), based on some features of an animal (represented by \(x\)).
- One solution is to have an algorithm that **tries to find a straight line (decision boundary)** to separate the elephants and dogs.
Classification

• Then, to classify a new animal as either an elephant or a dog, it checks on which side of the decision boundary it falls, and makes its prediction accordingly.
Classification

• This type of learning algorithm **models the conditional distribution of y given x:**
  \[ p(y \mid x) \]

• It involves **learning** \( p(y \mid x) \) **directly or from the mapping** from input to the output variables from the training data set.
Classification

• Algorithms that learn \( p(y \mid x) \) directly or from the input-output mapping is called \textbf{discriminative learning algorithms}.
• Example: logistic regression, perceptron algorithm.

There is an \textbf{alternative approach} for learning classification
Classification

• An alternative approach is:
• First, looking at elephants, we can build a model of what elephants look like.
Classification

• Then, looking at dogs, we can build a separate model of what dogs look like.
Finally, to classify a new animal, we can match the new animal against the elephant & dog model, and classify.
Classification

- This type of algorithm tries to \textbf{model} the features \((x)\) and the class \((y)\).
- I.e., it models the \textbf{joint distribution} \(p(x, y)\).
- According to the \textbf{product rule of probability}:
  \[
  p(x, y) = p(x \mid y)p(y)
  \]
- Hence, this type of algorithm computes \(p(x \mid y)\) and \(p(y)\) to model the joint distribution \(p(x, y)\).
Classification

- These algorithms that model $p(x, y)$ are called **generative learning algorithms**.

- For instance, if $y$ indicates whether an example is a dog (0) or an elephant (1), then $p(x \mid y = 0)$ models the distribution of dogs’ features, and $p(x \mid y = 1)$ models the distribution of elephants’ features.

\[
p(x, y) = p(x \mid y)p(y)
\]
Classification: Generative Model

- Then, after modeling \( p(y) \) (called the class priors) and \( p(x \mid y) \), the generative algorithm uses Bayes’ rule to derive the posterior distribution on \( y \) given \( x \):

\[
p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)}
\]

Here, \( p(x) = p(x \mid y = 1)p(y = 1) + p(x \mid y = 0)p(y = 0) \)

Thus the denominator can also be expressed in terms of the quantities \( p(x \mid y) \) and \( p(y) \) that we have learned.
Classification: Generative Model

• If we were calculating $p(y \mid x)$ in order to make a prediction, then we don’t actually need to calculate the denominator.

• Because in a Maximum a Posteriori (MAP) estimate we maximize $p(y \mid x)$ with respect to $y$.

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)}$$

$$y_{MAP} = \arg\max_y p(y \mid x)$$

$$y_{MAP} = \arg\max_y \frac{p(x \mid y)p(y)}{p(x)}$$

$$y_{MAP} = \arg\max_y p(x \mid y)p(y)$$
Classification: Generative Model

• Recall that if we ignore the prior distribution or assume a uniform prior, the MAP reduces to MLE.
Classification: Generative Model

- Therefore, in a **generative model** for classification we estimate $p(y \mid x)$ by:
  - **Model the sample** via likelihood $p(x \mid y)$
  - **Model the prior** $p(y)$
  
- Then using Bayes’ rule find posterior: $p(y \mid x) \sim p(x \mid y) p(y)$
  
- The MAP estimate gives **optimal classifier**.
  
- Hence, $y_{\text{MAP}}$ is a **Bayes’ optimal classifier**.

\[
y_{\text{MAP}} = \arg\max_y p(x \mid y)p(y)
\]

Bayes’ Optimal Classifier: pick the **most confident class**!
Classification: Generative Model

• We will explore the **Generative Model** of learning by using **Bayes’ optimal classifier**.

\[ y_{MAP} = \underset{y}{\text{argmax}} \ p(x \mid y)p(y) \]
Bayes’ Classifier

- Let’s build a Bayes’ optimal classifier.
- We would like to model $p(Y = y | X = x)$, where $X$ is a feature vector and $Y$ is its associated class level.
- We want to predict the success level of a student.
- Our training data has $n$ rows, $C$ classes and $d$ features (binary $d = 2$).

$Y = y \in \{1, 2, ..., C\}$

$\hat{X} = \hat{x} \in \{0, 1\}^d$
Classification

- How many parameters do we require to compute \( p(Y = y \mid X = x) \)?
- Note: \( p(Y = y \mid X = x) = p(X = x \mid Y = y)p(Y = y) \)
- Prior \( p(Y = y) \): \((C - 1)\) parameters
- Likelihood \( p(X = x \mid Y = y) \): \((2^d - 1)C\)
- Total no. of parameters = \((C - 1) + (2^d - 1)C\)
- \(O(2^d)\): Lots of parameters!

Curse of dimensionality, high variance & overfitting!

\[
Y = y \in \{1, 2, \ldots, C\}
\]

\[
\hat{X} = \hat{x} \in \{0, 1\}^d
\]

Each feature can take any one of the two states \{0/1\}, for each class
Naïve Bayes Classifier

• As a remedy to the overfitting and potential curse of dimensionality, we make a very strong assumption about the likelihood \( p(x_1, x_2, \ldots, x_d \mid y) \).

• Assume that the \( x_j \)'s (features) are conditionally independent given \( y \).

\[
p(x_1, \ldots, x_d \mid y) = p(x_1 \mid y)p(x_2 \mid y) \ldots p(x_d \mid y) = \prod_{j=1}^{d} p(x_j \mid y)
\]

The conditional independence assumption is our inductive bias.
Naïve Bayes Classifier

• This assumption is called the Naïve Bayes (NB) assumption.

• The resulting algorithm is called the Naïve Bayes classifier.

\[
p(x_1, \ldots x_d \mid y) = p(x_1 \mid y)p(x_2 \mid y) \ldots p(x_d \mid y) = \prod_{j=1}^{d} p(x_j \mid y)
\]

Important to note:
However, the conditional independence doesn’t mean absolute independence:

\[
p(x_1 \mid x_2) = p(x_1)
\]
Naïve Bayes Classifier

• Why does the conditional independence assumption work (in some domains, such as text classification)?
• Consider classifying emails into spam and ham.
• Words like “lottery”, “prize” and “win” are all likely indicators that the email might be spam.
• On the other hand words like “Bayes”, “Learning”, “Classifier” are good indicators of a ham email.
Naïve Bayes Classifier

• If we know that the email is spam, then we don’t need, for example, the word “win” to explain the existence of the word “lottery”.
  • $p(x_1 = \text{win}, x_2 = \text{lottery} | y = \text{spam}) = p(x_1 = \text{win} | y = \text{spam}) \ p(x_2 = \text{lottery} | y = \text{spam})$

• The word “win” is conditionally independent from “lottery” given the email is spam.
Naïve Bayes Classifier

- Thus, we can model the probability of occurrence for each of these words, given the respective class and then use it to score the likelihood of a text.

\[ p(x_1, x_2, x_3 \mid y) = p(x_1 \mid y) p(x_2 \mid y) p(x_3 \mid y) \]
Naïve Bayes Classifier

• When **does not** the conditional independence assumption work?
• It doesn’t work in some domains, such as *image classification*.
• Consider a handwritten digit image recognition problem.

Say that we model an image as a **vector of pixel values**.

But single **pixels are not independently generated**, depending only on the label.
Naïve Bayes Classifier

• For example, if we look at a **black pixel (grayscale value 0)** at one of the corners of the image, the pixels around are also **very likely to be black**.

Thus, the inductive assumption about the conditional independence **will not hold in these scenarios**.

\[
p(x_1, x_2, x_3 \mid y) \neq p(x_1 \mid y) p(x_2 \mid y) p(x_3 \mid y)
\]
Naïve Bayes Classifier

• How does the NB assumption reduce the complexity of parameter space (hence potential overfitting)?

• Due to the NB assumption, how many parameters do we require to compute \( p(Y = y | X = x) \)?

\[
p(x_1, \ldots, x_d | y) = p(x_1 | y)p(x_2 | y) \ldots p(x_d | y) = \prod_{j=1}^{d} p(x_j | y)
\]
Naïve Bayes Classifier

- How many parameters do we require to compute \( p(Y = y | X = x) \)?
- Prior \( p(Y = y) \): \((C - 1)\) parameters
- Likelihood \( p(x_1, \ldots, x_d | Y = y) = p(x_1 | y) \cdots p(x_d | y) \): \(d(2^1 - 1)C = dC\)
- Total no. of parameters: \((C - 1) + dC\)
- Its complexity is reduced to \(O(dC)\).

Each feature can take any one of the two states \(\{0/1\}\), for each class.

Hence it’s relatively immune to overfitting.

\[
p(x_1, \ldots, x_d | y) = p(x_1 | y)p(x_2 | y) \cdots p(x_d | y) = \prod_{j=1}^{d} p(x_j | y)
\]
Naïve Bayes Classifier

• To better understand the impact of the **NB assumption**, consider an **email classification** problem.
• An email \( x \) is represented by **3 features**.
• It can be classified as either spam or not-spam (\( y = 0/1 \)).
• **How many parameters** do we require to compute the likelihood \( p(X = x \mid Y = y) \)?
• Let’s calculate \( p(x_1, x_2, x_3 \mid y) \).

\[
\begin{align*}
x & \text{ represents an email} \\
x = \{x_1, x_2, x_3\} \\
x_1 & = \text{Viagra} \\
x_2 & = \text{Lottery} \\
x_3 & = \text{Sky} \\
y & = \{0, 1\}
\end{align*}
\]
**Naïve Bayes Classifier**

- The **full joint distribution** table for \( p(x_1, x_2, x_3, y) \) requires to have \( 2 \times 2 \times 2 \times 2 = 2^4 = 16 \) entries
Naïve Bayes Classifier

- Using the **chain rule of probability**:
  - \( p(x_1, x_2, x_3 \mid y) = p(x_1 \mid x_2, x_3, y) \cdot p(x_2, x_3 \mid y) \)
  - \( p(x_1, x_2, x_3 \mid y) = p(x_1 \mid x_2, x_3, y) \cdot p(x_2 \mid x_3, y) \cdot p(x_3 \mid y) \)

<table>
<thead>
<tr>
<th>( p(x_1 \mid x_2, x_3, y) )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( p(x_2 \mid x_3, y) )</th>
<th>( x_3 )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( p(x_3 \mid y) )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

# Probabilities = 7
# Probabilities = 3
# Probabilities = 1
Naïve Bayes Classifier

• Now, let’s use the conditional independence assumption and calculate the number of parameters needed to determine the likelihood $p(x_1, x_2, x_3 \mid y)$.

• **NB assumption**: $p(x_1, x_2, x_3 \mid y) = p(x_1 \mid y) p(x_2 \mid y) p(x_3 \mid y)$

$x$: represents an email

$x = \{x_1, x_2, x_3\}$

$x_1 = \text{Viagra}$

$x_2 = \text{Lottery}$

$x_3 = \text{Sky}$

$y = \{0, 1\}$
Naïve Bayes Classifier: Bayes Classifier vs. Naïve Bayes Classifier

$p(x_1, x_2, x_3 \mid y) = p(x_1 \mid x_2, x_3, y) \cdot p(x_2 \mid x_3, y) \cdot p(x_3 \mid y)$

# Probabilities = 7

<table>
<thead>
<tr>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

# Probabilities = 1

<table>
<thead>
<tr>
<th>$x_3$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

# Probabilities = 3

<table>
<thead>
<tr>
<th>$x_3$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

# Probabilities = 1

**NB assumption**: $p(x_1, x_2, x_3 \mid y) = p(x_1 \mid y) \cdot p(x_2 \mid y) \cdot p(x_3 \mid y)$

<table>
<thead>
<tr>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

# Probabilities = 1

<table>
<thead>
<tr>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

# Probabilities = 1

<table>
<thead>
<tr>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

# Probabilities = 1
Naïve Bayes Classifier

- Carefully note the **difference** between Bayes Classifier and Naïve Bayes Classifier.

\[
y_{MAP} = \arg\max_y p(y \mid \vec{x})
\]
\[
= \arg\max_y p(y)p(\vec{x} \mid y)
\]

**Bayes Classifier**

\[O(2^d)\]

**Naïve Bayes Classifier**

\[
y_{MAP} = \arg\max_y p(y \mid \vec{x})
\]
\[
= \arg\max_y p(y)\prod_{j=1}^{d} p(x_j \mid y)
\]

\[O(dC)\]
Naïve Bayes Classifier

• Why is the Naïve Bayes model so-called?

• The model is called “naïve” because we do not expect the features to be independent, even conditional on the class label.

• We note that even though the Naïve Bayes assumption is an extremely strong assumptions, the resulting algorithm works well on many problems (e.g., text classification).

\[
p(x_1, x_2, x_3 \mid y) = p(x_1 \mid y) p(x_2 \mid y) p(x_3 \mid y)
\]
Naïve Bayes Classifier

- So, how does a NB classifier **make prediction**?
- First of all, note that it’s a **generative model**.
- Hence, we form a **joint model of a d-dimensional feature vector** \( X \) and the corresponding class label \( Y = y \):

\[
p(\tilde{X}, Y = y) = p(\tilde{X} \mid Y = y)p(Y = y)
\]

Due to **NB assumption** (conditional independence of \( X \) given \( Y = y \))

\[
p(\tilde{X}, Y = y) = p(Y = y) \prod_{j=1}^{d} p(x_j \mid Y = y)
\]
Naïve Bayes Classifier

• Then, we use Bayes’ rule to form a classifier for a novel input $x$:

$$p(Y = y | \hat{x}) = \frac{p(\hat{x} | Y = y)p(Y = y)}{p(\hat{x})}$$

$$p(Y = y | \hat{x}) = \frac{p(\hat{x} | Y = y)p(Y = y)}{\sum_y p(\hat{x} | Y = y)p(Y = y)}$$

Let’s look at a simple example of a classification problem
Naïve Bayes Classifier: Example

• Say that a market research team wants to classify the radio station listeners into two groups: the “young” and “old”.
• Say that the variable “r” represents the radio stations: \( r_1, r_2, r_3, r_4 \)
• Each of the variables \( r_1, r_2, r_3, r_4 \) can take the states like or dislike.
• The variable “age” can take the value: young or old.
Naïve Bayes Classifier: Example

• Assume that, given the knowledge that a customer is either “young” or “old”, this is sufficient to determine whether or not a customer will like a particular radio station, independent of their likes or dislikes for any other stations:

\[ p(r_1, r_2, r_3, r_4 \mid \text{age} ) = p(r_1 \mid \text{age} ) p(r_2 \mid \text{age} ) p(r_3 \mid \text{age} ) p(r_4 \mid \text{age} ) \]
Naïve Bayes Classifier: Example

- Thus the information about the **age** of the customer *determines the individual radio station preferences* without needing to know anything else.

- See the **problem specification**:

<table>
<thead>
<tr>
<th>Young listener: chance of liking a station</th>
<th>Old listener: chance of liking a station</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio 1: 95%</td>
<td>Radio 1: 3%</td>
</tr>
<tr>
<td>Radio 2: 5%</td>
<td>Radio 2: 82%</td>
</tr>
<tr>
<td>Radio 3: 2%</td>
<td>Radio 3: 34%</td>
</tr>
<tr>
<td>Radio 4: 20%</td>
<td>Radio 4: 92%</td>
</tr>
</tbody>
</table>

90% of the listeners are old: $p(\text{old}) = 0.9$
Naïve Bayes Classifier: Example

• Now the **problem statement**:  
• Say that a **new customer likes** Radio1, and Radio3, but dislikes Radio2 and Radio4.  
• What is the probability that the **new customer is young**?  
• In other words, determine:

\[
p(\text{age} = \text{young} \mid r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike})
\]

<table>
<thead>
<tr>
<th><strong>Young listener: chance of liking a station</strong></th>
<th><strong>Old listener: chance of liking a station</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio 1: 95%</td>
<td>Radio 1: 3%</td>
</tr>
<tr>
<td>Radio 2: 5%</td>
<td>Radio 2: 82%</td>
</tr>
<tr>
<td>Radio 3: 2%</td>
<td>Radio 3: 34%</td>
</tr>
<tr>
<td>Radio 4: 20%</td>
<td>Radio 4: 92%</td>
</tr>
</tbody>
</table>
Naïve Bayes Classifier: Example

- We use the Bayes’ rule, then the NB classifier model.

\[
p(\text{age} = \text{young} \mid r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike})
\]

\[
= \frac{p(r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} \mid \text{age} = \text{young})p(\text{age} = \text{young})}{\sum_{\text{age}} p(r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} \mid \text{age})p(\text{age})}
\]

\[
p(Y = y \mid \tilde{x}) = \frac{p(\tilde{x} \mid Y = y)p(Y = y)}{\sum_{y} p(\tilde{x} \mid Y = y)p(Y = y)}
\]
Naïve Bayes Classifier: Example

- Due to the **NB assumption**, the **likelihood** can be written:

\[
p( r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} \mid \text{age} = \text{young} )
\]

\[
= p( r_1 = \text{like} \mid \text{young} ) p( r_2 = \text{dislike} \mid \text{young} ) p( r_3 = \text{like} \mid \text{young} ) p( r_4 = \text{dislike} \mid \text{young} )
\]

\[
p(\text{age} = \text{young} \mid r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} )
\]

\[
= \frac{ p( r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} \mid \text{age} = \text{young} ) p(\text{age} = \text{young} )}{ \sum_{\text{age}} p( r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} \mid \text{age} ) p(\text{age}) }
\]
Naïve Bayes Classifier: Example

Therefore, using the NB assumption in the numerator:

\[
p(age = \text{young} \mid r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike})
\]

\[
= \frac{p(r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} \mid age = \text{young})p(age = \text{young})}{\sum_{age} p(r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} \mid age) p(age)}
\]

\[
= \frac{p(r_1 = \text{like} \mid \text{young})p(r_2 = \text{dislike} \mid \text{young})p(r_3 = \text{like} \mid \text{young})p(r_4 = \text{dislike} \mid \text{young})p(age = \text{young})}{\sum_{age} p(r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} \mid age) p(age)}
\]
Naïve Bayes Classifier: Example

- Therefore, using the NB assumption in the numerator:

\[
p(\text{age} = \text{young} \mid r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike})
\]

\[
= \frac{p(r_1 = \text{like} \mid \text{young})p(r_2 = \text{dislike} \mid \text{young})p(r_3 = \text{like} \mid \text{young})p(r_4 = \text{dislike} \mid \text{young})p(\text{age} = \text{young})}{\sum_{\text{age}} p(r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} \mid \text{age})p(\text{age})}
\]

- **Young listener: chance of liking a station**
  - Radio 1: 95%
  - Radio 2: 5%
  - Radio 3: 2%
  - Radio 4: 20%

- **Old listener: chance of liking a station**
  - Radio 1: 3%
  - Radio 2: 82%
  - Radio 3: 34%
  - Radio 4: 92%

- 90% of the listeners are old: \(p(\text{old}) = 0.9\)

- \(p(\text{young}) = 0.1\)

- \(p(r_1 = \text{like} \mid \text{young}) = 0.95\)
- \(p(r_2 = \text{dislike} \mid \text{young}) = 0.95\)
- \(p(r_3 = \text{like} \mid \text{young}) = 0.02\)
- \(p(r_4 = \text{dislike} \mid \text{young}) = 0.8\)
Naïve Bayes Classifier: Example

- Compute the numerator:

\[
p(r_1 = \text{like} \mid \text{young}) p(r_2 = \text{dislike} \mid \text{young}) p(r_3 = \text{like} \mid \text{young}) p(r_4 = \text{dislike} \mid \text{young}) p(\text{age} = \text{young})
\]

Numerator: \(0.95 \times 0.95 \times 0.02 \times 0.8 \times 0.1 = 0.0014\)

- Young listener: chance of liking a station
  - Radio 1: 95%
  - Radio 2: 5%
  - Radio 3: 2%
  - Radio 4: 20%

- Old listener: chance of liking a station
  - Radio 1: 3%
  - Radio 2: 82%
  - Radio 3: 34%
  - Radio 4: 92%

90% of the listeners are old: \(p(\text{old}) = 0.9\)

\[
p(r_1 = \text{like} \mid \text{young}) = 0.95
\]
\[
p(r_2 = \text{dislike} \mid \text{young}) = 0.95
\]
\[
p(r_3 = \text{like} \mid \text{young}) = 0.02
\]
\[
p(r_4 = \text{dislike} \mid \text{young}) = 0.8
\]
\[
p(\text{young}) = 0.1
\]

\[
\frac{p(r_1 = \text{like} \mid \text{young}) p(r_2 = \text{dislike} \mid \text{young}) p(r_3 = \text{like} \mid \text{young}) p(r_4 = \text{dislike} \mid \text{young}) p(\text{age} = \text{young})}{\sum_{\text{age}} p(r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} \mid \text{age}) p(\text{age})}
\]
Naïve Bayes Classifier: Example

• Compute the denominator:
• The denominator is equal to the sum of the value of the numerator and the corresponding term evaluated assuming the customer is old:

\[
\sum_{\text{age}} p( r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} | \text{age} ) p(\text{age})
\]

\[
p(r_1 = \text{like} | \text{young}) p( r_2 = \text{dislike} | \text{young}) p( r_3 = \text{like} | \text{young}) p( r_4 = \text{dislike} | \text{young}) p(\text{age} = \text{young})
\]

\[+\]

\[
p(r_1 = \text{like} | \text{old}) p( r_2 = \text{dislike} | \text{old}) p( r_3 = \text{like} | \text{old}) p( r_4 = \text{dislike} | \text{old}) p(\text{age} = \text{old})
\]

\[
= \frac{p(r_1 = \text{like} | \text{young}) p( r_2 = \text{dislike} | \text{young}) p( r_3 = \text{like} | \text{young}) p( r_4 = \text{dislike} | \text{young}) p(\text{age} = \text{young})}{\sum_{\text{age}} p( r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} | \text{age} ) p(\text{age})}
\]
Naïve Bayes Classifier: Example

- Compute the denominator (another part assuming age = old):

\[
p(r_1 = \text{like} \mid \text{old})p(r_2 = \text{dislike} \mid \text{old})p(r_3 = \text{like} \mid \text{old})p(r_4 = \text{dislike} \mid \text{old})p(\text{age} = \text{old})
\]

\[
0.03 \times 0.18 \times 0.66 \times 0.08 \times 0.9 = 1.3219 \times 10^{-4}
\]

Young listener: chance of liking a station
- Radio 1: 95%
- Radio 2: 5%
- Radio 3: 2%
- Radio 4: 20%

Old listener: chance of liking a station
- Radio 1: 3%
- Radio 2: 82%
- Radio 3: 34%
- Radio 4: 92%

90% of the listeners are old: \(p(\text{old}) = 0.9\)

\[
p(\text{r}_1 = \text{like} \mid \text{old}) = 0.03
\]
\[
p(\text{r}_2 = \text{dislike} \mid \text{old}) = 0.18
\]
\[
p(\text{r}_3 = \text{like} \mid \text{old}) = 0.66
\]
\[
p(\text{r}_4 = \text{dislike} \mid \text{old}) = 0.08
\]

\[
p(\text{old}) = 0.9
\]
Naïve Bayes Classifier: Example

- Compute the denominator:
- The denominator is equal to the sum of the value of the numerator and the corresponding term evaluated assuming the customer is old,

\[
\sum_{\text{age}} p(r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} \mid \text{age}) p(\text{age})
\]

\[
p(r_1 = \text{like} \mid \text{young})p(r_2 = \text{dislike} \mid \text{young}) p(r_3 = \text{like} \mid \text{young}) p(r_4 = \text{dislike} \mid \text{young})p(\text{age} = \text{young})
\]

+ 

\[
p(r_1 = \text{like} \mid \text{old})p(r_2 = \text{dislike} \mid \text{old}) p(r_3 = \text{like} \mid \text{old}) p(r_4 = \text{dislike} \mid \text{old})p(\text{age} = \text{old})
\]

\[
= \frac{p(r_1 = \text{like} \mid \text{young})p(r_2 = \text{dislike} \mid \text{young}) p(r_3 = \text{like} \mid \text{young}) p(r_4 = \text{dislike} \mid \text{young})p(\text{age} = \text{young})}{\sum_{\text{age}} p(r_1 = \text{like}, r_2 = \text{dislike}, r_3 = \text{like}, r_4 = \text{dislike} \mid \text{age}) p(\text{age})}
\]
Naïve Bayes Classifier: Example

• Finally:

\[
p(age = young \mid r_1 = like, r_2 = dislike, r_3 = like, r_4 = dislike) = \frac{p(r_1 = like \mid young)p(r_2 = dislike \mid young)p(r_3 = like \mid young)p(r_4 = dislike \mid young)p(age = young)}{\sum_{age} p(r_1 = like, r_2 = dislike, r_3 = like, r_4 = dislike \mid age)p(age)}
\]

\[
= \frac{0.0014}{0.0014 + 1.3219 \times 10^{-4}} = 0.9161
\]

There is a 91.61% chance of the customer being young!
Simplicity of Naïve Bayes Classifiers: Benefit and Bane
Naïve Bayes Classifier

• For the rest of our journey with the Naïve Bayes classifiers, we will have **three main goals**.
  - Understand the **simplicity** (both conceptual and computational) of the Naïve Bayes algorithm.
  
  - **Benefit** of this simplicity.

  - **Bane** (limitation) of this simplicity.
Naïve Bayes Classifier

- The Naïve Bayes algorithm is fueled by two “simplicity”.
  - Simple inductive bias: conditional independence
  - Simple computation: statistical estimation of the prior and likelihood
Naïve Bayes Classifier

• Because of its simplicity, Naïve Bayes was a very popular choice for building spam filters during 80s and 90s.

• However, for classifying images it is not as effective.

• The main reason is the inadequacy of its simplicity!

• The inductive bias (conditional independence assumption) doesn’t hold in many domains (e.g., image classification).
Naïve Bayes Classifier

• It is due to this incorrect assumption about the “world” Naïve Bayes classifiers lost its popularity.

• Advanced models such as Artificial Neural Network was invented for image classification.

For an empirical understanding see my Github repository:
https://github.com/rhasanbd/Naive-Bayes-Classifier-Benefit-Bane
Naïve Bayes Classifier

• Before we pursue the route towards advanced classifiers, we will investigate why the Naïve Bayes classifiers excel in some domains, such as text classification.

• We will make a foray into text classification problem and understand why the Naïve Bayes classifiers are so successful in text classification.

For an empirical understanding see my Github repository: https://github.com/rhasanbd/Naive-Bayes-Algorithms-Foray-Into-Text-Classification