Classification – Naïve Bayes

Frequentist Learning

Categorical Features: Multivariate Bernoulli Distribution

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

• Categorical Features: Binary Valued
• Multivariate Bernoulli Distribution
• Frequentist Learning Approach
• Multivariate Bernoulli NB: Model Fitting
• Maximum Likelihood Estimation (MLE)
• Limitation of MLE
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
Naïve Bayes Classifier

- To ground our discussion into practical scenarios, let’s consider a document classification problem.
- Consider an example of a set of emails that consists of both spam and non-spam (or ham) emails.
- We want to automatically classify a new email pertaining to either spam or ham.

<table>
<thead>
<tr>
<th>Index</th>
<th>Document</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“toy”, “kid”, “toy”</td>
<td>Ham</td>
</tr>
<tr>
<td>2</td>
<td>“lottery”, “congratulations”, “lottery”, “toy”</td>
<td>Spam</td>
</tr>
<tr>
<td>3</td>
<td>“congratulations”, “paper”, “accepted”, “win”</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>“lottery”, “win”, “prize”, “win”</td>
<td>Spam</td>
</tr>
</tbody>
</table>
Naïve Bayes Classifier: Text Classification

• We search through the set of spam and ham emails and find the most commonly occurring words or terms (not including so-called ‘stop words’ such as ‘a’ or ‘the’).
• Let’s say that we have found $d$ number of unique terms.
• Using these terms we create a vocabulary or dictionary.

Vocabulary = [“toy”, “kid”, “lottery”, “congratulations”, “paper”, “accepted”, “win”, “prize”]
Naïve Bayes Classifier: Text Classification

• The terms in the Vocabulary are called **features**.
• We **represent an email (document)** via a **feature vector**.
• Its **length** is equal to the number of terms in the vocabulary/dictionary.

<table>
<thead>
<tr>
<th>Index</th>
<th>Document</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“toy”, “kid”, “toy”</td>
<td>Ham</td>
</tr>
<tr>
<td>2</td>
<td>“lottery”, “congratulations”, “lottery”, “toy”</td>
<td>Spam</td>
</tr>
<tr>
<td>3</td>
<td>“congratulations”, “paper”, “accepted”, “win”</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>“lottery”, “win”, “prize”, “win”</td>
<td>Spam</td>
</tr>
</tbody>
</table>

**Vocabulary** = [“toy”, “kid”, “lottery”, “congratulations”, “paper”, “accepted”, “win”, “prize”]
Naïve Bayes Classifier: Text Classification

• Thus, each email is represented by a **d-dimensional vector** of features.
• This is called the “**bag of words**” representation.
• This is a **crude representation** of the document since it **discards word order**.

<table>
<thead>
<tr>
<th>Index</th>
<th>Document</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“toy”, “kid”, “toy”</td>
<td>Ham</td>
</tr>
<tr>
<td>2</td>
<td>“lottery”, “congratulations”, “lottery”, “toy”</td>
<td>Spam</td>
</tr>
<tr>
<td>3</td>
<td>“congratulations”, “paper”, “accepted”, “win”</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>“lottery”, “win”, “prize”, “win”</td>
<td>Spam</td>
</tr>
</tbody>
</table>
Naïve Bayes Classifier

- Each sample in the training data (i.e., document) is represented as a feature vector $\mathbf{x} = [x_1 \ldots x_d]$
- For example, the features are the terms (words) in the vocabulary.
- The number of features is $d$.
- The number of possible values of each feature is $M$: $\mathbf{x} \in \{0, \ldots, M\}^d$
- Here $M$ denotes the number of terms in a particular document.

### Vocabulary

- [“toy”, “kid”, “lottery”, “congratulations”, “paper”, “accepted”, “win”, “prize”]

### Document 1: $M = 3$

<table>
<thead>
<tr>
<th>Index</th>
<th>Document</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“toy”, “kid”, “toy”</td>
<td>Ham</td>
</tr>
<tr>
<td>2</td>
<td>“lottery”, “congratulations”, “lottery”, “toy”</td>
<td>Spam</td>
</tr>
<tr>
<td>3</td>
<td>“congratulations”, “paper”, “accepted”, “win”</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>“lottery”, “win”, “prize”, “win”</td>
<td>Spam</td>
</tr>
</tbody>
</table>

Toy: 2
Kid: 1
Lottery: 0
Congratulations: 0
Etc.
Naïve Bayes Classifier

• In the feature vector \( \tilde{x} = [x_1 \ldots x_d] \), each \( x \) is a random variable.

• The random variable \( y \) represents the class of the sample (e.g., Ham or Spam).

• It can be one of \( C \) discrete states:

\[
y = \{y_1, y_2, y_3, \ldots, y_C\}
\]

<table>
<thead>
<tr>
<th>Index</th>
<th>Document</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“toy”, “kid”, “toy”</td>
<td>Ham</td>
</tr>
<tr>
<td>2</td>
<td>“lottery”, “congratulations”, “lottery”, “toy”</td>
<td>Spam</td>
</tr>
<tr>
<td>3</td>
<td>“congratulations”, “paper”, “accepted”, “win”</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>“lottery”, “win”, “prize”, “win”</td>
<td>Spam</td>
</tr>
</tbody>
</table>
Naïve Bayes Classifier

- We have \( N \) training examples.
  \[
  D = \{ (\vec{x}_1, y_1), \ldots, (\vec{x}_N, y_N) \}
  \]
- Our goal is to **classify the feature vector** \( \vec{x} \in \{0, 1, \ldots, M\}^d \)
- In other words, we want to determine the **conditional probability** of the class \( y = c \) given the feature vector \( \vec{x} \):
  \[
  p(y = c \mid \vec{x})
  \]
- Out of all possible states of \( y \), we want to pick the state that **maximizes the probability**.

<table>
<thead>
<tr>
<th>Index</th>
<th>Document</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“toy”, “kid”, “toy”</td>
<td>Ham</td>
</tr>
<tr>
<td>2</td>
<td>“lottery”, “congratulations”, “lottery”, “toy”</td>
<td>Spam</td>
</tr>
<tr>
<td>3</td>
<td>“congratulations”, “paper”, “accepted”, “win”</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>“lottery”, “win”, “prize”, “win”</td>
<td>Spam</td>
</tr>
</tbody>
</table>

**Classify document 1:**
\[
\vec{x} = \{“toy”, “kid”\}
\]
\[
p(y = Ham \mid \vec{x})
\]
\[
p(y = Spam \mid \vec{x})
\]
Naïve Bayes Classifier

• For this classification problem we will use a generative approach:
  
  \[ p(\mathbf{x}, y = c) = p(y = c)p(\mathbf{x} \mid y = c) \]

• We compute \( p(y = c) \) and \( p(\mathbf{x} \mid y = c) \).

• Then, using the Bayes’ rule, we get the conditional probability of the class \( y = c \) given the feature vector \( \mathbf{x} \):
  
  \[ p(y = c \mid \mathbf{x}) \propto p(y = c)p(\mathbf{x} \mid y = c) \]

• Applying the NB assumption (i.e., the features are conditionally independent given the class):
  
  \[ p(y = c \mid \mathbf{x}) \propto p(y = c) \prod_{j=1}^{d} p(x_j \mid y = c) \]
Naïve Bayes Classifier

• Carefully note the key aspect of the NB algorithm.
• Our goal is to learn the discriminative function \( p(y = c \mid \tilde{x}) \)
• To do this, we model the sample & class probabilities.

\[
p(y = c \mid \tilde{x}) \propto p(y = c)p(\tilde{x} \mid y = c)
\]

Thus, NB algorithm is rich as it enables us to model (reproduce) the reality!
Naïve Bayes Classifier

- \[ p(y = c \mid \vec{x}) \propto p(y = c) \prod_{j=1}^{d} p(x_j \mid y = c) \]
- \( p(y = c) \): prior probability distribution of the class (e.g., number of documents belonging to a class)
- It will be denoted by \( \pi_c \)
- \( p(x_j \mid y) \): likelihood of a feature component given the class
- It will be denoted by \( \theta_{jc} \)
- Here \( \theta_{jc} \) is the probability that feature \( j \) occurs in class \( c \).
- Note that \( \vec{\theta} \) is a d-dimensional vector: for every feature it gives the probability of the appearance of that feature in a class.
Naïve Bayes Classifier

• Thus the **posterior** distribution for a single sample $\tilde{x}_i$ (e.g., document, indexed by $i$) is $p(y_i = c \mid \tilde{x}_i)$:

$$p(y_i = c \mid \tilde{x}_i) = p(y_i) \prod_{j=1}^{d} p(x_{ij} \mid y_i = c)$$

$$p(y_i = c \mid \tilde{x}_i, \tilde{\pi}, \tilde{\theta}) = \pi_c \prod_{j=1}^{d} \theta_{jc}$$

Our goal is to **learn the parameters** $\theta_{jc}$ & $\pi_c$
Naïve Bayes Classifier

- We will discuss two types of features:
  - Categorial Features
  - Real-Valued Features
Naïve Bayes Classifier

• For **categorical** features, we will consider **two possible cases**:
  
  • **Case 1**: Feature values are Boolean (binary valued): $\tilde{x} \in \{0, 1\}^d$
  
  • For example, it captures whether a word **occurs** in the document (1) or not (0).
  
  • **Case 2**: Feature values are K-dimensional categorical: $\tilde{x} \in \{0, \ldots, M\}^d$
  
  • For example, feature values are **count of feature occurrences**, showing **how many times** the corresponding word occurs in the document.
Naïve Bayes Classifier

- For categorial features, we will consider **two possible cases**:

<table>
<thead>
<tr>
<th>Index</th>
<th>Document</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“toy”, “kid”, “toy”</td>
<td>Ham</td>
</tr>
<tr>
<td>2</td>
<td>“lottery”, “congratulations”, “lottery”, “toy”</td>
<td>Spam</td>
</tr>
<tr>
<td>3</td>
<td>“congratulations”, “paper”, “accepted”, “win”</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>“lottery”, “win”, “prize”, “win”</td>
<td>Spam</td>
</tr>
</tbody>
</table>

**Feature: “toy” & “lottery” in Document 1**

**Case 1** (Boolean Features):
- “toy” = 1
- “lottery” = 0

**Case 2** (Multi-valued Categorical):
- “toy” = 2
- “lottery” = 0
Naïve Bayes Classifier

- **Real-Valued Features**
  - Feature values are continuous (real-valued) numbers.
Naïve Bayes Classifier: Model Fitting

• How do we “train” a naïve Bayes classifier?
• There are two approaches for the training.
  - Frequentist Learning Approach
  - Bayesian Learning Approach

The Bayesian Learning approach prevents overfitting.

We can show that the Frequentist model is a special case of the Bayesian model.
Naïve Bayes Classifier: Model Fitting

• There are two approaches for the training.
  - Frequentist Learning Approach (MLE)
  - Bayesian Learning Approach (MAP)

First we will discuss the
Frequentist Learning Approach
Frequentist Learning Approach
Frequentist Learning Approach

• In the Frequentist approach, we **construct the likelihood function** for the data.
• Then, we **maximize the likelihood** using MLE.
• For constructing the likelihood function we need to **construct the probability distribution** for two model parameters:
  • $\theta_j$: Feature Likelihood
  • $\pi_c$: Class Prior

\[
p(y_i = c | \tilde{x}_i) = p(y_i) \prod_{j=1}^{d} p(x_{ij} | y_i = c)
\]

\[
p(y_i = c | \tilde{x}_i, \bar{\pi}, \bar{\theta}) = \pi_c \prod_{j=1}^{d} \theta_{jc}
\]
For $N$ documents, class prior probability distribution is similar to the multiple coin toss scenario.

We need to determine: out of $N$ documents (e.g., $N$ coin tosses), \#times class “c” occurred and \#times class “c” didn’t occur.

Thus we can represent the class prior probability by using a binomial distribution:

$$p(y_i = c) \propto (\pi_c)^{\sum_{i=1}^{N} \mathbb{1}(y_i=c)} (1 - \pi_c)^{\sum_{i=1}^{N} \mathbb{1}(y_i=\text{not } c)}$$

$$\sum_{i=1}^{N} \mathbb{1}(y_i = c)$$ number of samples in class c.

$$\sum_{i=1}^{N} \mathbb{1}(y_i = \text{not } c)$$: number of samples not in class c.
We consider whether in document $i$, the feature $j$ occurred or not.

It’s similar to a single coin toss in which either head/tail turns up.

Thus, feature likelihood $p(x_j \mid y = c)$ can be modeled as Bernoulli distribution for each class $c$ and each feature $j$:

$$p(x_j \mid y = c) = (\theta_{jc})^{x_j}(1 - \theta_{jc})^{1-x_j}$$
The multi-valued categorical feature, $x_{ij} \in \{0, 1, \ldots, M_i\}$, is similar to a side of a d-sided die.

For a document $i$, $M_i$ represents the total number of words (i.e., the total number of times the die is rolled).

We use $x_{ij}$ to denote the times feature $j$ occurred, (e.g., times the $j$ side of the die showed up).

Thus, the distribution of the multi-valued categorical features can be represented by the Multinomial distribution.

$$p(\bar{x}_i \mid y_i = c) = \prod_{j=1}^{d} p(x_{ij} \mid y_i = c) = \frac{M_i!}{\prod_{j=1}^{d} x_{ij}!} \prod_{j=1}^{d} (\theta_{jc})^{x_{ij}}$$
Feature likelihood:

$$p(y_i = c | \tilde{x}_i) = p(y_i) \prod_{j=1}^{d} p(x_{ij} | y_i = c)$$

Categorical Feature

Case 2: Multi-valued features

We consider in document \(i\) the **number of times** feature \(j\) occurred.

It’s similar to **multiple coin tosses** in which we count #times head occurred and #times it didn’t

For **document \(i\)**, that has \(M_i\) words, we use \(x_{ij}\) to denote the #times **feature \(j\)** occurred, and \((M_i - x_{ij})\) times it did not occur.

Thus, feature likelihood \(p(x_j | y = c)\) can be modeled as **binomial distribution** for each class \(c\) and each feature \(j\):

$$p(x_{ij} | y_i = c) \propto (\theta_{jc})^{x_{ij}} (1 - \theta_{jc})^{M_i - x_{ij}}$$
Assume that for each class $y = c$, the distribution of each continuous $x_j$ $p(x_j | y = c)$ follows a **Gaussian distribution**.

For each document $i$, the Gaussian distribution is defined by the **mean and standard deviation** specific to $x_{ij}$ and $y_i = c$:

$$p(x_{ij} | y_i = c, \mu_{jc}, \sigma_{jc}^2) = \frac{1}{\sqrt{2\pi\sigma_{jc}^2}}e^{-\frac{1}{2\sigma_{jc}^2}(x_{ij} - \mu_{jc})^2}$$

To compute feature likelihood, we need to **estimate** the mean and standard deviation of each of these Gaussians:

$$\hat{\mu}_{jc} = \hat{\sigma}_{jc}^2 =$$
Class Prior Probability Distribution:

**Binomial distribution:**
\[ p(y_i = c) \propto (\pi_c) \sum_{i=1}^{N} \mathbb{1}(y_i = c) (1 - \pi_c) \sum_{i=1}^{N} \mathbb{1}(y_i = \text{not } c) \]

Feature likelihood:

**Bernoulli distribution:**
\[ p(x_j \mid y = c) = (\theta_{jc})^{x_j} (1 - \theta_{jc})^{1-x_j} \]

**Categorical Feature**

Case 1: Binary-valued features

Case 2: Multi-valued features

**Gaussian distribution:**
\[ p(x_{ij} \mid y_i = c, \mu_{jc}, \sigma_{jc}^2) = \frac{1}{\sqrt{2\pi\sigma_{jc}^2}} e^{-\frac{1}{2\sigma_{jc}^2}(x_{ij} - \mu_{jc})^2} \]

**Multinomial distribution:**
\[ p(\tilde{x}_i \mid y_i = c) \propto \prod_{j=1}^{d} (\theta_{jc})^{x_{ij}} \]
Frequentist Learning Approach

- Categorical Features are Binary Valued: Multivariate Bernoulli NB model
- Categorical Features are Multi-valued: Multinomial NB model
- Real-valued Features: Gaussian NB model
Frequentist Learning Approach

- Categorical Features are Binary Valued: Multivariate Bernoulli NB model
- Categorical Features are Multi-valued: Multinomial NB model
- Real-valued Features: Gaussian NB model
Multivariate Bernoulli NB

- We will cover the case of **binary features** in detail.
- Models for other feature types can then be generalized.
- The binary feature, $x_j \in \{0, 1\}$, is similar to **coin toss scenario**.
- We consider whether the feature $x_j$ **occurs in a document or not** (like whether head or tail occurs in a **single** coin toss).
- Thus we use the **Bernoulli distribution** to model the distribution of the features.

\[
p(\tilde{x} | y = c, \tilde{\theta}) = \prod_{j=1}^{d} \text{Bernoulli}(x_j | \theta_{jc})
\]

Here $\theta_{jc}$ is the probability that feature $j$ occurs in class $c$. 
Multivariate Bernoulli NB

- For each feature, we need to estimate the values for two classes:
  \[ p(x_j = 1 \mid y = c) := \theta_{jc} \]
- It represents the probability that feature \( j \) occurs in class \( c \).

Note that we don’t care about the frequency of the terms

<table>
<thead>
<tr>
<th>Index</th>
<th>Document</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“toy”, “kid”, “toy”</td>
<td>Ham</td>
</tr>
<tr>
<td>2</td>
<td>“lottery”, “congratulations”, “lottery”, “toy”</td>
<td>Spam</td>
</tr>
<tr>
<td>3</td>
<td>“congratulations”, “paper”, “accepted”, “win”</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>“lottery”, “win”, “prize”, “win”</td>
<td>Spam</td>
</tr>
</tbody>
</table>
Multivariate Bernoulli NB

- The other probability that feature j doesn’t occur in class c $p(x_j = 0 \mid y = c)$ is given by the normalization requirement:

$$p(x_j = 0 \mid y = c) = 1 - p(x_j = 1 \mid y = c) = 1 - \theta_{jc}$$

- $p(x_j = 1 \mid y = c) := \theta_{jc}$
- $p(x_j = 0 \mid y = c) := 1 - \theta_{jc}$
Multivariate Bernoulli NB

- Now we need to **construct the probability distribution** for two model parameters:
  - $\theta_{jc}$: Feature Likelihood
  - $\pi_c$: Class Prior

\[
p(y_i = c \mid \tilde{x}_i) = p(y_i) \prod_{j=1}^{d} p(x_{ij} \mid y_i = c)
\]

\[
p(y_i = c \mid \tilde{x}_i, \tilde{\pi}, \tilde{\theta}) = \pi_c \prod_{j=1}^{d} \theta_{jc}
\]

\[
p(x_j = 1 \mid y = c) := \theta_{jc}
\]

\[
p(x_j = 0 \mid y = c) := 1 - \theta_{jc}
\]
Multivariate Bernoulli NB

- Since features are binary valued, we can compare their likelihood with the single coin toss scenario.
- In a single document we only consider whether a feature component occurs or not (just like whether head/tail shows up in single coin toss).
- Thus, feature likelihood $p(x_j \mid y = c)$ can be modeled as Bernoulli distribution for each class $c$ and each feature $j$:

$\begin{align*}
    p(x_j \mid y = c) &= \text{Bernoulli}(x_j \mid \theta_{jc}) \\
    p(x_j \mid y = c) &= (\theta_{jc})^{x_j} (1 - \theta_{jc})^{1-x_j}
\end{align*}$
Multivariate
Bernoulli NB

• Feature likelihood \( p(x_j \mid y = c) \):
  \[
p(x_j \mid y = c) = \text{Bernoulli}(x_j \mid \theta_{jc})
  \]
  \[
p(x_j \mid y = c) = (\theta_{jc})^{x_j} (1 - \theta_{jc})^{1-x_j}
  \]

• For a feature vector of sample \( i \) with \textbf{d components} \( p(\vec{x}_i \mid y_i = c) \):
  \[
p(\vec{x}_i \mid y_i = c) = \prod_{j=1}^{d} p(x_{ij} \mid y_i = c) = \prod_{j=1}^{d} (\theta_{jc_i})^{x_{ij}} (1 - \theta_{jc_i})^{1-x_{ij}}
  \]

- Now we need to \textbf{model the probability distribution} for two model parameters:
  - \( \theta_{jc} \): Feature Likelihood
  - \( \pi_c \): Class Prior
Multivariate Bernoulli NB

- Recall the random variable $y$ represents the class of the sample, which can be one of $C$ discrete states:
  \[ y = \{y_1, y_2, y_3, \ldots, y_C\} \]
- How do we model the distribution of $y$?
- The states of $y$ is similar to the value of the sides of a die.

Consider that the die has $C$ sides.

Each side represents a value of $y$.

To generate a value of $y$, we roll the C-sided die.
Prior: Multinomial Distribution

• When we roll a C-sided die, what is the probability that a side will turn up?
• We can model the outcome of a C-sided die toss by using multinomial variables.

To model the probability of multinomial variables, we use a Multinoulli distribution (die rolled once).

It is a generalization of Bernoulli distribution (coin tossed once).
Multivariate Bernoulli NB

• Thus the **class probability** is modeled using a **multinoulli distribution**:

\[
p(y_i = c) = Multinoulli(y_i = c \mid \bar{\pi}) = \prod_{c=1}^{C} \pi_c \mathbb{I}(y_i = c)
\]

Here \( \mathbb{I} \) is an **indicator function**

\[
\mathbb{I}(y_i = c) = \begin{cases} 
1 & \text{if } y_i = c \\
0 & \text{if } y_i \neq c 
\end{cases}
\]

\( \pi_c \) represents the **probability of the occurrence of the class “c”** (e.g., the “c” side of the die turns up)
Multivariate Bernoulli NB

• Multinoulli distribution:

\[ p(y_i = c) = \text{Multinoulli}(y_i = c \mid \vec{\pi}) = \prod_{c=1}^{C} \pi_c^{\mathbb{I}(y_i=c)} \]

• If we **roll the die multiple times** then we represent the probability of a class using **multinominal distribution**.

• Thus for **N samples**:

\[
\prod_{i=1}^{N} p(y_i = c) = \text{Multinomial}(y_i = c \mid \vec{\pi}) = \prod_{i=1}^{N} \prod_{c=1}^{C} \pi_c^{\mathbb{I}(y_i=c)}
\]
Multivariate Bernoulli NB

- $\prod_{i=1}^{N} p(y_i = c) = \text{Multinomial}(y_i = c \mid \vec{\pi}) = \prod_{i=1}^{N} \prod_{c=1}^{C} \pi_{c} \mathbb{I}(y_i=c)$

- Note that the **individual components** of a multinomial random vector $\pi_c$ (individual class or side of a die) are binomial and have a **binomial distribution**:

$$p(y_i = c) \propto (\pi_c) \sum_{i=1}^{N} \mathbb{I}(y_i=c) \cdot (1 - \pi_c) \sum_{i=1}^{N} \mathbb{I}(y_i=not \ c)$$

Binomial Distribution for **coin toss**: for $n$ trials head occurs $k$ times

$$\text{Bin}(k \mid n, \theta) = \binom{n}{k} \theta^k (1 - \theta)^{n-k}$$

- $\sum_{i=1}^{N} \mathbb{I}(y_i = c)$: number of **samples in class c**.
- $\sum_{i=1}^{N} \mathbb{I}(y_i = not \ c)$: number of **samples not in class c**.

It’s binomial because we count the no. of times class “c” occurred in N samples & the no. of times it **didn’t occur**.
• Thus, we have **modeled the probability distribution** for two model parameters:
  • $\theta_{jc}$: Feature Likelihood
  • $\pi_c$: Class Prior

Feature likelihood $p(x_j | y = c)$ can be modeled as **Bernoulli distribution** for each class $c$ and each feature $j$:

$$p(x_j | y = c) = \text{Bernoulli}(x_j | \theta_{jc})$$

$$p(x_j | y = c) = (\theta_{jc})^{x_j} (1 - \theta_{jc})^{1-x_j}$$

Note that the **individual components** of a multinomial random vector $\pi_c$ (individual class or side of a die) are binomial and have a **binomial distribution**:

$$p(y_i = c) \propto (\pi_c)^{\sum_{i=1}^{N} 1(y_i=c)} (1 - \pi_c)^{\sum_{i=1}^{N} 1(y_i=\text{not } c)}$$
Multivariate Bernoulli NB: Model Fitting
Multivariate Bernoulli NB: Model Fitting

- Assuming the i.i.d. assumption holds, we apply **MLE to learn the optimal parameters**:
  - The class prior distribution: \( p(y = c) = p(c) = \pi_c \)
  - The class-dependent feature distributions \( p(x_j | c) = \theta_{jc} \)

\[
\begin{align*}
p(x_j = 1 | y = c) & := \theta_{jc} \\
p(x_j = 0 | y = c) & := 1 - \theta_{jc} \\
p(y = c) & := \pi_c
\end{align*}
\]
Multivariate Bernoulli NB: Model Fitting

• **Frequentist Learning Approach:**
  - Consider a **dataset** \( D = \{ (\tilde{x}_i, y_i), i = 1, 2, \ldots, N \} \) containing feature vectors \( \tilde{x}_i \) and class labels \( y_i \).

\[
D = \{ (\tilde{x}_1, y_1), \ldots, (\tilde{x}_N, y_N) \}
\]

- Using MLE we **learn** the parameters \( \pi_c \) and \( \theta_{jc} \) that **maximize the likelihood** of the dataset \( D \)
- First we present a **summary** of the calculation.
Multivariate Bernoulli NB: Model Fitting

- **Frequentist Learning Approach:**
  
  Using MLE we learn the optimal parameters $\pi_c$ and $\theta_{jc}$

  \[
  \hat{\pi}_c = p(y = c) = \frac{N_c}{N} = \frac{\text{Number of samples belonging to class } c}{\text{Total number of samples}}
  \]

  \[
  \hat{\theta}_{jc} = p(x_j = 1 \mid y = c) = \frac{N_{jc}}{N_c} = \frac{\text{Number of samples belonging to class } c \text{ that contain feature } j}{\text{Number of samples belonging to class } c}
  \]

  Then, the **optimal NB classifier** finds the class probability of a features as:

  \[
p(y = c \mid x_j) = p(y = c) \ p(x_j \mid y = c) = \pi_c \theta_{jc}
  \]
Multivariate Bernoulli NB: Classification Boundary

- After determining the optimal NB classifier model, we can classify a novel input $x^*$ as class 1 if:

\[ p(c = 1|x^*) > p(c = 0|x^*) \]

Using MLE we will derive these expressions.

Then, the optimal NB classifier finds the class probability of a features as:

\[ p(y = c | x_j) = p(x_j | y = c)p(y = c) = \pi_c \theta_{jc} \]
Multivariate Bernoulli
NB: Model Fitting

- The likelihood of all (N) data points (features and class labels):

\[ D = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_N, y_N)\} \]

\[ L(D) = p(D) = \prod_{i=1}^{N} p(\vec{x}_i, y_i = c) \]

Product rule

\[ p(D) = \prod_{i=1}^{N} p(y_i)p(\vec{x}_i | y_i = c) \]

NB assumption

\[ p(D) = \prod_{i=1}^{N} \left[ p(y_i) \prod_{j=1}^{d} p(x_{ij} | y_i = c) \right] \]

For all C classes

\[ p(D) = \prod_{i=1}^{N} \left[ \sum_{c=1}^{C} p(y_i = c) \mathbb{I}(y_i = c) \prod_{j=1}^{d} \sum_{c=1}^{C} p(x_{ij} | y_i = c) \right] \]

Here \( \mathbb{I} \) is an indicator function

\[ \mathbb{I}(y_i = c) = \begin{cases} 1 & \text{if } y_i = c \\ 0 & \text{if } y_i \neq c \end{cases} \]
Multivariate Bernoulli NB: Model Fitting

- The likelihood of all (N) data points (features and class labels):

$$L(D) = p(D) = \prod_{i=1}^{N} p(x_i, y_i = c)$$

$$p(D) = \prod_{i=1}^{N} \left[ \prod_{c=1}^{C} p(y_i = c)^{\mathbb{1}(y_i = c)} \prod_{j=1}^{d} \prod_{c=1}^{C} p(x_{ij} | y_i = c) \right]$$

$$p(x_i | y = c, \hat{\theta}) = \prod_{j=1}^{d} \text{Bernoulli}(x_j | \theta_{jc})$$

$$p(x_i | y = c) = \prod_{j=1}^{d} p(x_{ij} | y_i = c) = \prod_{j=1}^{d} (\theta_{jc})^{x_{ij}} (1 - \theta_{jc})^{1-x_{ij}}$$

$$p(D | \pi, \hat{\theta}) = \prod_{i=1}^{N} \left[ \prod_{c=1}^{C} \pi_{c}^{\mathbb{1}(y_i = c)} \prod_{j=1}^{d} \prod_{c=1}^{C} (\theta_{jc})^{x_{ij}} (1 - \theta_{jc})^{1-x_{ij}} \right]$$
Multivariate Bernoulli NB: Model Fitting

- The **log-likelihood** is given by:

\[
p(D \mid \vec{\pi}, \vec{\theta}) = \prod_{i=1}^{N} \left[ \prod_{c=1}^{C} \pi_c^{(y_i=c)} \prod_{j=1}^{d} \prod_{c=1}^{C} (\theta_{jc})^{x_{ij}} (1 - \theta_{jc})^{1-x_{ij}} \right]
\]

\[
log p(D \mid \vec{\pi}, \vec{\theta}) = \log \prod_{i=1}^{N} \prod_{c=1}^{C} \pi_c^{(y_i=c)} + \log \prod_{i=1}^{N} \prod_{j=1}^{d} \prod_{c=1}^{C} (\theta_{jc})^{x_{ij}} (1 - \theta_{jc})^{1-x_{ij}}
\]

\[
log p(D \mid \vec{\pi}, \vec{\theta}) = \sum_{i=1}^{N} \sum_{c=1}^{C} \log \pi_c^{(y_i=c)} + \log \prod_{i=1}^{N} \prod_{j=1}^{d} \prod_{c=1}^{C} (\theta_{jc})^{x_{ij}} (1 - \theta_{jc})^{1-x_{ij}}
\]
Multivariate Bernoulli NB: Model Fitting

• The **log-likelihood** is given by:

\[
\log p(D | \vec{\pi}, \vec{\theta}) = \sum_{i=1}^{N} \sum_{c=1}^{C} \log \pi_c^{\mathbb{1}(y_i=c)} + \log \prod_{i=1}^{N} \prod_{j=1}^{d} \prod_{c=1}^{C} (\theta_{jc})^{x_{ij}} (1 - \theta_{jc})^{1-x_{ij}}
\]

We write the log-likelihood as:

\[
\log p(D | \vec{\pi}, \vec{\theta}) = \log L(D | \vec{\pi}) + \log L(D | \vec{\theta})
\]

\[
\log L(D | \vec{\pi}) = \sum_{i=1}^{N} \sum_{c=1}^{C} \log \pi_c^{\mathbb{1}(y_i=c)}
\]

\[
\log L(D | \vec{\theta}) = \log \prod_{i=1}^{N} \prod_{j=1}^{d} \prod_{c=1}^{C} (\theta_{jc})^{x_{ij}} (1 - \theta_{jc})^{1-x_{ij}}
\]
Multivariate Bernoulli NB: Model Fitting

\[ \log L(D | \hat{\pi}) = \sum_{i=1}^{N} \sum_{c=1}^{C} \log \pi_c \mathbb{I}(y_i = c) \]

\[ \log p(D | \hat{\pi}) = \sum_{i=1}^{N} \sum_{c=1}^{C} \log \pi_c \mathbb{I}(y_i = c) \]

Consider the 1\textsuperscript{st} Term:

\[ \log L(D | \hat{\theta}) = \log \prod_{i=1}^{N} \prod_{j=1}^{d} \prod_{c=1}^{C} (\theta_{jc})^{x_{ij}} (1 - \theta_{jc})^{1-x_{ij}} \]

\[ \log p(D | \hat{\theta}) = \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{I}(y_i = c) \log \pi_c \]
Multivariate Bernoulli NB: Model Fitting

Consider the 1st Term:

$$\log p(D | \vec{\pi}, \vec{\theta}) = \log L(D | \vec{\pi}) + \log L(D | \vec{\theta})$$

$$\log p(D | \vec{\pi}) = \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{I}(y_i = c) \log \pi_c$$

Split it into two terms by considering the following

Note that the individual components of a multinomial random vector $\pi_c$ are binomial and have a binomial distribution:

$$p(y_i = c) \propto (\pi_c)^{\sum_{i=1}^{N} \mathbb{I}(y_i = c)} (1 - \pi_c)^{\sum_{i=1}^{N} \mathbb{I}(y_i = not c)}$$

$$\mathbb{I}(y_i = c) = 1: \text{if the sample is in class } c.$$  
$$\mathbb{I}(y_i = not c) = 1: \text{if the sample is not in class } c.$$
Multivariate Bernoulli NB: Model Fitting

Consider the 1st Term:

\[ \log p(D | \tilde{\pi}, \tilde{\theta}) = \log L(D | \tilde{\pi}) + \log L(D | \tilde{\theta}) \]

\[ \log p(D | \tilde{\pi}) = \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{I}(y_i = c) \log \pi_c \]

\[ \log p(D | \tilde{\pi}) = N_c \log \pi_c + (N - N_c) \log(1 - \pi_c) \]

\[ N_c = \sum_{i=1}^{N} \mathbb{I}(y_i = c) \]

\[ N - N_c \] is the number of samples not in class c.

\[ N_c \] is the number of samples in class c.
Multivariate Bernoulli NB: Model Fitting

Consider the 1st Term:

\[ \log p(D | \vec{\pi}, \vec{\theta}) = \log L(D | \vec{\pi}) + \log L(D | \vec{\theta}) \]

\[ \log p(D | \vec{\pi}) = N_c \log \pi_c + (N - N_c) \log (1 - \pi_c) \]

To maximize the likelihood we take the derivative of \( \log p(D | \vec{\pi}) \) wrt \( \pi_c \), and set it to 0:

\[ \frac{\partial (\log p(D | \vec{\pi}))}{\partial \pi_c} = 0 \]
Multivariate Bernoulli NB: Model Fitting

Consider the 1\textsuperscript{st} Term:

\[
\partial \left( \log p(D \mid \pi) \right) \over \partial \pi_c = 0
\]

Thus the MLE for the class prior:

\[
\hat{\pi}_c = \frac{N_c}{N}
\]

\[
\log p(D \mid \hat{\pi}, \hat{\theta}) = \log L(D \mid \pi) + \log L(D \mid \hat{\theta})
\]

\[
\log p(D \mid \pi) = N_c \log \pi_c + (N - N_c) \log(1 - \pi_c)
\]

\[
\frac{N_c}{\pi_c} + \frac{N - N_c}{1 - \pi_c} \cdot (-1) = 0
\]

\[
\Rightarrow \frac{N_c}{\pi_c} = \frac{N - N_c}{1 - \pi_c}
\]

\[
\Rightarrow N_c - N_c \pi_c = N \pi_c - N_c \pi_c
\]

\[
\Rightarrow N \pi_c = N_c
\]

\[
\text{Number of samples belonging to class } c = \frac{N_c}{N}
\]

\[
\text{Total number of samples}
\]
Multivariate Bernoulli NB: Class Prior

- Let’s use an example to illustrate how to calculate class prior probabilities by using the derived formula.

\[
\hat{\pi}_c = \frac{N_c}{N}
\]

\[
= \frac{\text{Number of samples belonging to class } c}{\text{Total number of samples}}
\]

\[
\hat{\pi}_{\text{Spam}} = \frac{N_{\text{Spam}}}{N}
\]

\[
\hat{\pi}_{\text{Ham}} = \frac{N_{\text{Ham}}}{N}
\]

\[
\hat{\pi}_{\text{Spam}} = \frac{2}{4} = \frac{1}{2}
\]

\[
\hat{\pi}_{\text{Ham}} = \frac{2}{4} = \frac{1}{2}
\]

<table>
<thead>
<tr>
<th>Index</th>
<th>Document</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“toy”, “kid”, “toy”</td>
<td>Ham</td>
</tr>
<tr>
<td>2</td>
<td>“lottery”, “congratulations”, “lottery”, “toy”</td>
<td>Spam</td>
</tr>
<tr>
<td>3</td>
<td>“congratulations”, “paper”, “accepted”, “win”</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>“lottery”, “win”, “prize”, “win”</td>
<td>Spam</td>
</tr>
</tbody>
</table>
Multivariate Bernoulli NB: Model Fitting

Consider the 1st Term:

\[ \pi, \hat{\theta} \]

To compute \( \pi_c \) we can use an elegant technique.

Thus, it’s a constrained optimization problem subject to an equality constraint.

We can use the method of Lagrange multipliers.

\[
\log p(D | \pi, \hat{\theta}) = \log L(D | \pi) + \log L(D | \hat{\theta})
\]

\[
\log p(D | \pi) = \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{I}(y_i = c) \log \pi_c
\]

Observe that while optimizing \( \log p(D | \pi) \) we need to respect the following constraint:

\[
\sum_{c=1}^{C} \pi_c = 1
\]
Multivariate Bernoulli NB: Model Fitting

Consider the 1\textsuperscript{st} Term:

\[
\log p(D | \vec{\pi}, \vec{\theta}) = \log L(D | \vec{\pi}) + \log L(D | \vec{\theta})
\]

The \textbf{method of Lagrange multipliers}

\[
\log p(D | \vec{\pi}) = \sum_{i=1}^{N} \sum_{c=1}^{C} \text{I}(y_i = c) \log \pi_c
\]

\[
\log p(D | \vec{\pi}) = \sum_{c=1}^{C} N_c \log \pi_c
\]

\[N_c \text{ is the number of examples in class } c.\]

We want to maximize the likelihood \( \log p(D | \vec{\pi}) \) \textbf{subject to the constraint:}

\[
\sum_{c=1}^{C} \pi_c = 1
\]

\[N_c = \sum_{i=1}^{N} \text{I}(y_i = c)\]
Multivariate Bernoulli NB: Model Fitting

Consider the 1st Term:

We define the unconstrained objective function or the Lagrangian:

\[ \mathcal{L} = \sum_{c=1}^{C} N_c \log \pi_c + \lambda \left( 1 - \sum_{c=1}^{C} \pi_c \right) \]

Here \( \lambda \) is the Lagrange multiplier

Taking partial derivative with respect to \( \pi_c \) and \( \lambda \), and setting it to 0:

\[ \frac{\partial \mathcal{L}}{\partial \pi_c} = 0 \]

\[ \frac{\partial \mathcal{L}}{\partial \lambda} = 0 \]
Multivariate Bernoulli NB: Model Fitting

Lagrangian of the 1st Term:

\[
\frac{\partial \mathcal{L}}{\partial \lambda} = 0
\]

\[
\frac{\partial \mathcal{L}}{\partial \pi_c} = 0
\]

\[
1 - \sum_{c=1}^{C} \pi_c = 0
\]

\[
\Rightarrow \sum_{c=1}^{C} \pi_c = 1
\]

\[
\frac{N_c}{\pi_c} - \lambda = 0
\]

\[
\Rightarrow \frac{N_c}{\pi_c} = \lambda
\]

\[
\Rightarrow N_c = \lambda \pi_c
\]

Taking the sum on both sides:

\[
N_c = \lambda \pi_c
\]

\[
\Rightarrow \sum_{c=1}^{C} N_c = \lambda \sum_{c=1}^{C} \pi_c
\]

\[
\Rightarrow N = \lambda
\]

\[
N_c = N \pi_c
\]

\[
\hat{\pi}_c = \frac{N_c}{N}
\]
Multivariate Bernoulli NB: Model Fitting

\[ \log p(D \mid \vec{\pi}, \vec{\theta}) = \log L(D \mid \vec{\pi}) + \log L(D \mid \vec{\theta}) \]

\[ \log L(D \mid \vec{\pi}) = \sum_{i=1}^{N} \sum_{c=1}^{C} \log \pi_c^{y_i=c} \]

Consider the 2\textsuperscript{nd} Term:

\[ \log L(D \mid \vec{\theta}) = \log \prod_{i=1}^{N} \prod_{j=1}^{d} \prod_{c=1}^{C} (\theta_{jc})^{x_{ij}} (1 - \theta_{jc})^{1-x_{ij}} \]

Product of N terms

\[ \log p(D \mid \vec{\theta}) = \log \prod_{c=1}^{C} \prod_{j=1}^{d} (\theta_{jc})^{x_{1j} \cdot (1 - \theta_{jc})^{1-x_{1j}}} \cdot \prod_{j=1}^{d} (\theta_{jc})^{x_{2j} \cdot (1 - \theta_{jc})^{1-x_{2j}}} \ldots \prod_{j=1}^{d} (\theta_{jc})^{x_{Nj} \cdot (1 - \theta_{jc})^{1-x_{Nj}}} \]
Multivariate Bernoulli NB: Model Fitting

Consider the 2\textsuperscript{nd} Term:

\[
\log p(D \mid \hat{\pi}, \hat{\theta}) = \log L(D \mid \hat{\pi}) + \log L(D \mid \hat{\theta})
\]

\[
\log p(D \mid \hat{\theta}) = \log \prod_{c=1}^{C} \left[ \prod_{j=1}^{d} \left( \theta_{jc} \right)^{x_{1j}} \left( 1 - \theta_{jc} \right)^{1-x_{1j}} \right] \prod_{j=1}^{d} \left( \theta_{jc} \right)^{x_{2j}} \left( 1 - \theta_{jc} \right)^{1-x_{2j}} \cdots \prod_{j=1}^{d} \left( \theta_{jc} \right)^{x_{Nj}} \left( 1 - \theta_{jc} \right)^{1-x_{Nj}}
\]

\[
\log p(D \mid \hat{\theta}) = \log \prod_{c=1}^{C} \left[ \prod_{j=1}^{d} \left( \theta_{jc} \right)^{x_{1j} + \cdots + x_{Nj}} \left( 1 - \theta_{jc} \right)^{(1+\cdots+1)-(x_{1j}+\cdots+x_{Nj})} \right]
\]

\[
\log p(D \mid \hat{\theta}) = \log \prod_{c=1}^{C} \left[ \prod_{j=1}^{d} \left( \theta_{jc} \right)^{N_{jc}} \left( 1 - \theta_{jc} \right)^{N_c - N_{jc}} \right]
\]

\[N_c = \text{Number of documents belonging to class } c\]

\[N_{jc} = \text{Number of documents belonging to class } c \text{ that contain feature } j\]
Multivariate Bernoulli NB: Model Fitting

\[
\log p(D | \tilde{\theta}) = \log \prod_{c=1}^{c} \prod_{j=1}^{d} (\theta_{jc})^{N_{jc}} \cdot \prod_{j=1}^{d} (1 - \theta_{jc})^{N_{c} - N_{jc}}
\]

\(N_c = \text{Number of documents belonging to class } c\)

\(N_{jc}: \text{Number of documents belonging to class } c \text{ that contain feature } j\)

<table>
<thead>
<tr>
<th>Index</th>
<th>Document</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“toy”, “kid”, “toy”</td>
<td>Ham</td>
</tr>
<tr>
<td>2</td>
<td>“lottery”, “congratulations”, “lottery”, “toy”</td>
<td>Spam</td>
</tr>
<tr>
<td>3</td>
<td>“congratulations”, “paper”, “accepted”, “win”</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>“lottery”, “win”, “prize”, “win”</td>
<td>Spam</td>
</tr>
</tbody>
</table>

\(N_{\text{toy}, \text{Ham}} = 1\)
\(N_{\text{toy}, \text{Spam}} = 1\)

\(N_{\text{win}, \text{Ham}} = 1\)
\(N_{\text{win}, \text{Spam}} = 1\)

Note that we don’t care about the frequency of the terms
Multivariate Bernoulli NB: Model Fitting

Consider the 2nd Term:

\[ \log p(D | \tilde{\pi}, \tilde{\theta}) = \log L(D | \tilde{\pi}) + \log L(D | \tilde{\theta}) \]

\[ \log p(D | \tilde{\theta}) = \log \prod_{c=1}^{C} \left[ \prod_{j=1}^{d} (\theta_{jc})^{N_{jc}} \right] \cdot \prod_{j=1}^{d} \left( 1 - \theta_{jc} \right)^{N_c - N_{jc}} \]

\[ \log p(D | \tilde{\theta}) = \log \prod_{c=1}^{C} \left[ \prod_{j=1}^{d} (\theta_{jc})^{N_{jc}} \right] + \log \prod_{c=1}^{C} \left[ \prod_{j=1}^{d} (1 - \theta_{jc})^{N_c - N_{jc}} \right] \]

\[ \log p(D | \tilde{\theta}) = \sum_{c=1}^{C} \sum_{j=1}^{d} \log(\theta_{jc})^{N_{jc}} + \sum_{c=1}^{C} \sum_{j=1}^{d} \log(1 - \theta_{jc})^{N_c - N_{jc}} \]

\[ \log p(D | \tilde{\theta}) = \sum_{c=1}^{C} \sum_{j=1}^{d} N_{jc} \log(\theta_{jc}) + \sum_{c=1}^{C} \sum_{j=1}^{d} (N_c - N_{jc}) \log(1 - \theta_{jc}) \]
Multivariate Bernoulli NB: Model Fitting

Consider the 2\textsuperscript{nd} Term:

\[
\log p(D | \tilde{\theta}) = \sum_{c=1}^{C} \sum_{j=1}^{d} N_{jc} \log(\theta_{jc}) + \sum_{c=1}^{C} \sum_{j=1}^{d} (N_c - N_{jc}) \log(1 - \theta_{jc})
\]

To \textbf{maximize the likelihood} we take the derivative of \( \log p(D | \tilde{\theta}) \) \textbf{wrt} \( \theta_{jc} \), and set it to 0:

\[
\frac{\partial \log p(D | \tilde{\theta})}{\partial \theta_{jc}} = 0
\]

\[
\frac{N_{jc}}{\theta_{jc}} + \frac{N_c - N_{jc}}{1 - \theta_{jc}} (-1) = 0
\]

\[
\Rightarrow \frac{N_{jc}}{\theta_{jc}} = \frac{N_c - N_{jc}}{1 - \theta_{jc}}
\]

\[
\Rightarrow N_{jc} - N_{jc}\theta_{jc} = N_c\theta_{jc} - N_{jc}\theta_{jc}
\]

\[
\Rightarrow N_{jc} = N_{jc} \frac{N_{jc}}{N_c}
\]

\[\hat{\theta}_{jc} = \frac{N_{jc}}{N_c}\]
Multivariate Bernoulli NB: Model Fitting

$$\hat{\theta}_{jc} = \frac{N_{jc}}{N_c}$$

Consider the 2\textsuperscript{nd} Term:

$$logp(D | \vec{\pi}, \vec{\theta}) = logL(D | \vec{\pi}) + logL(D | \vec{\theta})$$

Note that we don’t care about the frequency of the terms

<table>
<thead>
<tr>
<th>Index</th>
<th>Document</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“toy”, “kid”, “toy”</td>
<td>Ham</td>
</tr>
<tr>
<td>2</td>
<td>“lottery”, “congratulations”, “lottery”, “toy”</td>
<td>Spam</td>
</tr>
<tr>
<td>3</td>
<td>“congratulations”, “paper”, “accepted”, “win”</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>“lottery”, “win”, “prize”, “win”</td>
<td>Spam</td>
</tr>
</tbody>
</table>

\[ \hat{\theta}_{toy,Ham} = \frac{N_{toy,Ham}}{N_{Ham}} = 1/2 \]

\[ \hat{\theta}_{toy,spam} = \frac{N_{toy,spam}}{N_{spam}} = 1/2 \]
Therefore, the optimal NB classifier finds the class probability of a feature as:

\[
p(y = c \mid x_j) = p(y = c) \cdot p(x_j \mid y = c) = \pi_c \theta_{jc}
\]

\[
\hat{\pi}_c = p(y = c) = \frac{N_c}{N} = \frac{\text{number of times class c occurs}}{\text{total number of datapoints}}
\]

\[
\hat{\theta}_{jc} = p(x_j = 1 \mid y = c) = \frac{N_{jc}}{N_c} = \frac{\text{number of times } x_j \text{ occurs for class c}}{\text{number of datapoints in class c}}
\]
Multivariate Bernoulli NB: Model Fitting

- **Frequentist Learning Approach:**

Using MLE we learn the optimal parameters $\pi_c$ and $\theta_{jc}$

\[
\hat{\pi}_c = p(y = c) = \frac{N_c}{N} = \frac{\text{Number of samples belonging to class } c}{\text{Total number of samples}}
\]

\[
\hat{\theta}_{jc} = p(x_j = 1 \mid y = c) = \frac{N_{jc}}{N_c} = \frac{\text{Number of samples belonging to class } c \text{ that contain feature } j}{\text{Number of samples belonging to class } c}
\]

Then, the **optimal NB classifier** finds the class probability of a features as:

\[
p(y = c \mid x_j) = p(y = c) \ p(x_j \mid y = c) = \pi_c \theta_{jc}
\]
Multivariate Bernoulli NB: Model Fitting

• Now that we have determined the NB classifier model, we can classify a novel input \( x^* \) as class 1 if

\[
p(c = 1|x^*) > p(c = 0|x^*)
\]
Naïve Bayes Classifier: Receiver Operating Characteristics (ROC) curve
Naïve Bayes Classifier: ROC Curve

• In binary classification, such as in designing spam filter, sometimes we need to make tradeoff between precision and recall.

• Thus, it is useful to create the Receiver Operating Characteristics (ROC) curve.

• We model the class decision making problem as follows by using a threshold $T$.

\[
p(y = c | \hat{x}) = \frac{p(\hat{x} | y = c)p(y = c)}{p(\hat{x})}
\]

\[
p(y = c | \hat{x}) = \frac{p(\hat{x} | y = c)p(y = c)}{\sum_y p(\hat{x} | y = c)p(y = c)} > T
\]
Naïve Bayes Classifier: ROC Curve

- By **varying** $T$ we can draw the Receiver Operating Characteristics (ROC) curve.
- The ROC curve helps to **choose a suitable** $T$ to classify more true negatives (correctly classified ham messages) at the expense of fewer true positives (correctly classified spam messages), or vice-versa.

\[
\frac{p(\hat{x} \mid Y = y_{spam})p(Y = y_{spam})}{\sum_{y} p(\hat{x} \mid Y = y_{spam})p(Y = y_{spam})} > T
\]
Multivariate NB Classifier: Example
Multivariate Bernoulli NB: Example

• Let’s look at an example of binary classification by using the Multivariate Bernoulli NB classifier.

\[
\frac{p(\hat{x} \mid Y = y_{spam})p(Y = y_{spam})}{\sum_{y} p(\hat{x} \mid Y = y_{spam})p(Y = y_{spam})} > T
\]
Multivariate Bernoulli NB: Example

• We have a set of labeled emails belonging to Spam and Ham.

<table>
<thead>
<tr>
<th>Index</th>
<th>Email</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“kid”, “money”, “paper”</td>
<td>Ham</td>
</tr>
<tr>
<td>2</td>
<td>“congratulations”, “kid”, “money”, “kid”, “kid”</td>
<td>Ham</td>
</tr>
<tr>
<td>3</td>
<td>“congratulations”, “prize”, “paper”, “paper”, “paper”</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>“congratulations”, “prize”</td>
<td>Ham</td>
</tr>
<tr>
<td>5</td>
<td>“prize”, “paper”, “paper”</td>
<td>Ham</td>
</tr>
<tr>
<td>6</td>
<td>“money”</td>
<td>Ham</td>
</tr>
</tbody>
</table>

We will train a Multivariate Bernoulli classifier to classify new emails.
Create the vocabulary

It has 5 terms

Thus, the **length** of the feature vector is 5

**Vocabulary** = [“congratulations”, “prize”, “kid”, “money” “paper”]

**Feature vector \( \mathbf{x} \):**

- \( x_1 \): congratulations
- \( x_2 \): prize
- \( x_3 \): kid
- \( x_4 \): money
- \( x_5 \): paper

<table>
<thead>
<tr>
<th>Index</th>
<th>Email</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“kid”, “money”, “paper”</td>
<td>Ham</td>
</tr>
<tr>
<td>2</td>
<td>“congratulations”, “kid”, “money”, “kid”, “kid”</td>
<td>Ham</td>
</tr>
<tr>
<td>3</td>
<td>“congratulations”, “prize”, “paper”, “paper”, “paper”</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>“congratulations”, “prize”</td>
<td>Ham</td>
</tr>
<tr>
<td>5</td>
<td>“prize”, “paper”, “paper”</td>
<td>Ham</td>
</tr>
<tr>
<td>6</td>
<td>“money”</td>
<td>Ham</td>
</tr>
<tr>
<td>7</td>
<td>“congratulations”, “paper”</td>
<td>Spam</td>
</tr>
<tr>
<td></td>
<td>“congratulations”, “money”, “congratulations”</td>
<td>Spam</td>
</tr>
<tr>
<td>8</td>
<td>“congratulations”, “prize”, “paper”, “congratulations”</td>
<td>Spam</td>
</tr>
<tr>
<td>9</td>
<td>“congratulations”, “prize”, “kid”, “money”, “money”</td>
<td>Spam</td>
</tr>
<tr>
<td>10</td>
<td>“congratulations”, “prize”, “money”, “money”</td>
<td>Spam</td>
</tr>
<tr>
<td>11</td>
<td>“congratulations”, “money”, “congratulations”, “prize”, “paper”</td>
<td>Spam</td>
</tr>
<tr>
<td>12</td>
<td>“congratulations”, “kid”, “money”, “congratulations”</td>
<td>Spam</td>
</tr>
<tr>
<td>13</td>
<td>“congratulations”, “congratulations”, “kid”</td>
<td>Spam</td>
</tr>
</tbody>
</table>
Represent the Ham and Spam emails using **binary-valued feature vectors**

**Vocabulary =**

[“congratulations”, “prize”, “kid”, “money” “paper”]

<table>
<thead>
<tr>
<th>Index</th>
<th>Email</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“kid”, “money”, “paper”</td>
<td>Ham</td>
</tr>
<tr>
<td>2</td>
<td>“congratulations”, “kid”, “money”, “kid”</td>
<td>Ham</td>
</tr>
<tr>
<td>3</td>
<td>“congratulations”, “prize”, “paper”, “paper”</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>“congratulations”, “prize”</td>
<td>Ham</td>
</tr>
<tr>
<td>5</td>
<td>“prize”, “paper”, “paper”</td>
<td>Ham</td>
</tr>
<tr>
<td>6</td>
<td>“money”</td>
<td>Ham</td>
</tr>
<tr>
<td>7</td>
<td>“congratulations”, “paper”</td>
<td>Spam</td>
</tr>
<tr>
<td>8</td>
<td>“congratulations”, “prize”, “paper”, “congratulations”</td>
<td>Spam</td>
</tr>
<tr>
<td>9</td>
<td>“congratulations”, “prize”, “kid”, “money”, “money”</td>
<td>Spam</td>
</tr>
<tr>
<td>10</td>
<td>“congratulations”, “prize”, “money”, “money”</td>
<td>Spam</td>
</tr>
<tr>
<td>11</td>
<td>“congratulations”, “money”, “congratulations”, “prize”, “paper”</td>
<td>Spam</td>
</tr>
<tr>
<td>12</td>
<td>“congratulations”, “kid”, “money”, “congratulations”</td>
<td>Spam</td>
</tr>
<tr>
<td>13</td>
<td>“congratulations”, “prize”, “kid”</td>
<td>Spam</td>
</tr>
</tbody>
</table>

Feature vector \( \mathbf{x} \):

- \( x_1 : \text{congratulations} \)
- \( x_2 : \text{prize} \)
- \( x_3 : \text{kid} \)
- \( x_4 : \text{money} \)
- \( x_5 : \text{paper} \)
Multivariate Bernoulli NB: Example

- Two tables are given for two classes: Ham & Spam
- Each column represents an email.
- The rows represent 5 attributes.

Class prior $p(\text{class})$: no. of times a class occurred in the data

$p(\text{Ham})$: $6/13$

$p(\text{Spam})$: $7/13$
Multivariate Bernoulli

NB: Example

• Compute the **likelihood of each component of the feature vector for the Ham**:

<table>
<thead>
<tr>
<th>H1</th>
<th>H2</th>
<th>H3</th>
<th>H4</th>
<th>H5</th>
<th>H6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Each row represents one component of the feature vector (e.g., row 1 : $x_1$)

**Ham**

- $\theta_{\text{congratulations,Ham}} = p(x_1 = 1 \mid \text{Ham}) = 3/6 = \frac{1}{2}$
- $\theta_{\text{prize,Ham}} = p(x_2 = 1 \mid \text{Ham}) = 3/6 = \frac{1}{2}$
- $\theta_{\text{kid,Ham}} = p(x_3 = 1 \mid \text{Ham}) = 2/6 = \frac{1}{3}$
- $\theta_{\text{money,Ham}} = p(x_4 = 1 \mid \text{Ham}) = 3/6 = \frac{1}{2}$
- $\theta_{\text{paper,Ham}} = p(x_5 = 1 \mid \text{Ham}) = 3/6 = \frac{1}{2}$

The likelihood of each component of the feature vector for the Ham can be calculated using the formula:

$$\hat{\theta}_{jc} = p(x_j = 1 \mid y = c) = \frac{N_{jc}}{N_c}$$

where $N_{jc}$ is the number of samples belonging to class $c$ that contain feature $j$, and $N_c$ is the number of samples belonging to class $c$. 
Multivariate Bernoulli
NB: Example

• Compute the likelihood of each component of the feature vector for the Spam:

<table>
<thead>
<tr>
<th>S7</th>
<th>S8</th>
<th>S9</th>
<th>S10</th>
<th>S11</th>
<th>S12</th>
<th>S13</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<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>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Spam

- \( \theta_{\text{congratulations}, \text{Spam}} = p(x_1 = 1 \mid \text{Spam}) = \frac{7}{7} = 1 \)
- \( \theta_{\text{prize}, \text{Spam}} = p(x_2 = 1 \mid \text{Spam}) = \frac{4}{7} \)
- \( \theta_{\text{kid}, \text{Spam}} = p(x_3 = 1 \mid \text{Spam}) = \frac{3}{7} \)
- \( \theta_{\text{money}, \text{Spam}} = p(x_4 = 1 \mid \text{Spam}) = \frac{5}{7} \)
- \( \theta_{\text{paper}, \text{Spam}} = p(x_5 = 1 \mid \text{Spam}) = \frac{3}{7} \)

Each row represents one component of the feature vector (e.g., row 1 : \( x_1 \))

\[ \hat{\theta}_{jc} = p(x_j = 1 \mid y = c) = \frac{N_{jc}}{N_c} = \frac{\text{Number of samples belonging to class c that contain feature } j}{\text{Number of samples belonging to class c}} \]
Multivariate Bernoulli NB: Example

- Let’s classify a new email.

New Email 1 = [“Congratulations”, “kid”, “money”, “money”, “Congratulations”]

New Email 1: \(x = (1, 0, 1, 1, 0)^T\)

Using Bayes’ rule, probability of being Spam:

\[
p(\text{Spam} | \vec{x}) = \frac{p(\vec{x} | \text{Spam})p(\text{Spam})}{p(\vec{x})} = \frac{p(\vec{x} | \text{Spam})p(\text{Spam})}{p(\vec{x} | \text{Spam})p(\text{Spam}) + p(\vec{x} | \text{Ham})p(\text{Ham})}
\]

How do we bring in the optimal NB classifier?
Multivariate Bernoulli

NB: Example

- Optimal NB classifier:
  1. Apply the **NB assumption**.
  2. Use the **MLE** to compute the prior $p(\text{Spam})$ and the likelihood $p(x \mid \text{Spam})$.

Applying NB assumption to the likelihood:

$$p(\text{Spam} \mid \hat{x}) = \frac{p(\hat{x} \mid \text{Spam})p(\text{Spam})}{p(\hat{x} \mid \text{Spam})p(\text{Spam}) + p(\hat{x} \mid \text{Ham})p(\text{Ham})}$$

Feature vector $\hat{x}$:
- $x_1$: congratulations
- $x_2$: prize
- $x_3$: kid
- $x_4$: money
- $x_5$: paper

$$p(\hat{x} \mid \text{Spam}) = p(x_1 \mid \text{Spam})p(x_2 \mid \text{Spam})p(x_3 \mid \text{Spam})p(x_4 \mid \text{Spam})p(x_5 \mid \text{Spam})$$
**Multivariate Bernoulli**

**NB: Example**

- **Optimal NB classifier:**
  
  \[
  \mathbf{x} = (1, 0, 1, 1, 0)^T
  \]

\[
p(\text{Spam} \mid \mathbf{x}) = \frac{p(x_1 \mid \text{Spam})p(x_2 \mid \text{Spam})p(x_3 \mid \text{Spam})p(x_4 \mid \text{Spam})p(x_5 \mid \text{Spam})p(\text{Spam})}{p(\mathbf{x} \mid \text{Spam})p(\text{Spam}) + p(\mathbf{x} \mid \text{Ham})p(\text{Ham})}
\]

\[
p(\text{Spam} \mid \mathbf{x}) = \frac{1 \times \frac{3}{7} \times \frac{3}{7} \times \frac{5}{7} \times \frac{4}{7} \times \frac{7}{13}}{\left[1 \times \frac{3}{7} \times \frac{3}{7} \times \frac{5}{7} \times \frac{4}{7} \times \frac{7}{13}\right] + \left[\frac{1}{2} \times \frac{1}{2} \times \frac{1}{3} \times \frac{1}{2} \times \frac{1}{2} \times \frac{6}{13}\right]} = 0.8076
\]

- For **Ham**:
  
  \[
p(x_1 = 1 \mid \text{Ham}) = 3/6 = \frac{1}{2} \\
p(x_2 = 1 \mid \text{Ham}) = 3/6 = \frac{1}{2} \\
p(x_3 = 1 \mid \text{Ham}) = 2/6 = \frac{1}{3} \\
p(x_4 = 1 \mid \text{Ham}) = 3/6 = \frac{1}{2} \\
p(x_5 = 1 \mid \text{Ham}) = 3/6 = \frac{1}{2}
\]

- For **Spam**:
  
  \[
p(x_1 = 1 \mid \text{Spam}) = 7/7 = 1 \\
p(x_2 = 1 \mid \text{Spam}) = 4/7 \\
p(x_3 = 1 \mid \text{Spam}) = 3/7 \\
p(x_4 = 1 \mid \text{Spam}) = 5/7 \\
p(x_5 = 1 \mid \text{Spam}) = 3/7
\]

\[
p(\text{Spam}) = 7/13 \\
p(\text{Ham}) = 6/13
\]

It’s > 0.5, hence the email is Spam
Multivariate Bernoulli

NB: Example

- Let’s consider another email for classification.

New Email 2 = [“prize”, “kid”, “money”, “paper”, “prize”, “money”, “money”]

New Email 2: \( x = (0, 1, 1, 1, 1)^T \)

In the training data, all spam emails has the word “congratulations”, which the new email doesn’t have.

What is the probability that this email is Spam \( p(\text{Spam} \mid x) \)?
Multivariate Bernoulli

NB: Example

- Optimal NB classifier: \( x = (0, 1, 1, 1, 1)^T \)

\[
p(Spam \mid \vec{x}) = \frac{p(x_1 \mid Spam)p(x_2 \mid Spam)p(x_3 \mid Spam)p(x_4 \mid Spam)p(x_5 \mid Spam)p(Spam)}{p(\vec{x} \mid Spam)p(Spam) + p(\vec{x} \mid Ham)p(Ham)}
\]

\[
p(Spam \mid \vec{x}) = \frac{0 \times \frac{4}{7} \times \frac{3}{7} \times \frac{5}{7} \times \frac{3}{7} \times \frac{7}{13} + \frac{1}{2} \times \frac{1}{2} \times \frac{1}{3} \times \frac{1}{2} \times \frac{1}{2} \times \frac{6}{13}}{0 \times \frac{4}{7} \times \frac{3}{7} \times \frac{5}{7} \times \frac{3}{7} \times \frac{7}{13} + \frac{1}{2} \times \frac{1}{2} \times \frac{1}{3} \times \frac{1}{2} \times \frac{1}{2} \times \frac{6}{13}} = 0
\]

**Ham**
- \( p(x_1 = 1 \mid Ham) = 3/6 = 1/2 \)
- \( p(x_2 = 1 \mid Ham) = 3/6 = 1/2 \)
- \( p(x_3 = 1 \mid Ham) = 2/6 = 1/3 \)
- \( p(x_4 = 1 \mid Ham) = 3/6 = 1/2 \)
- \( p(x_5 = 1 \mid Ham) = 3/6 = 1/2 \)

**Spam**
- \( p(x_1 = 1 \mid Spam) = 7/7 = 1 \)
- \( p(x_2 = 1 \mid Spam) = 4/7 \)
- \( p(x_3 = 1 \mid Spam) = 3/7 \)
- \( p(x_4 = 1 \mid Spam) = 5/7 \)
- \( p(x_5 = 1 \mid Spam) = 3/7 \)

\( p(Spam): 7/13 \)
\( p(Ham): 6/13 \)
Multivariate Bernoulli NB: Example

- We see that the email is **not Spam**: $p(\text{Spam} \mid x) = 0$.
- Our model makes an **extremely confident** classification!
- The model **overfits** the data!
- This is another example of the **black swan paradox**.

New Email 2 = [“prize”, “kid”, “money”, “paper”, “prize”, “money”, “money”]

<table>
<thead>
<tr>
<th>Index</th>
<th>Email</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>“congratulations”, “paper”, “congratulations”, “money”</td>
<td>Spam</td>
</tr>
<tr>
<td>8</td>
<td>“congratulations”, “prize”, “paper”, “congratulations”</td>
<td>Spam</td>
</tr>
<tr>
<td>9</td>
<td>“congratulations”, “prize”, “kid”, “money”, “money”</td>
<td>Spam</td>
</tr>
<tr>
<td>10</td>
<td>“congratulations”, “prize”, “money”, “money”</td>
<td>Spam</td>
</tr>
<tr>
<td>11</td>
<td>“congratulations”, “money”, “congratulations”, “prize”, “paper”</td>
<td>Spam</td>
</tr>
<tr>
<td>12</td>
<td>“congratulations”, “kid”, “money”, “congratulations”</td>
<td>Spam</td>
</tr>
<tr>
<td>13</td>
<td>“congratulations”, “congratulations”, “kid”</td>
<td>Spam</td>
</tr>
</tbody>
</table>
Multivariate Bernoulli

NB: Example

• What if we have a new email with a word ("lottery") that was not in the training data?

New Email 3 = ["lottery", "congratulations", "paper", "money"]

The likelihood of the new word "lottery" for Spam is 0:

\[ p("lottery" \mid \text{Spam}) = 0 \]

Although this new email is exactly similar to a Spam email (no. 7), our classifier will predict that it has 0 probability of being Spam.
Multivariate Bernoulli NB: Example

- This demonstrates a difficulty using the MLE with **sparse data**!
- One way to ameliorate this is to smooth the probabilities.
- For example, by adding a small number to the frequency counts to each attribute.

This ensures that there are **no zero probabilities** in the model.
Bayesian Naïve Bayes Classifier

• To **prevent overfitting** we introduce **extra parameters** ($\alpha$ and $\beta$) to **add small numbers** to the frequency counts as follows.

\[
\bar{\pi}_c = \frac{\alpha_c + N_c}{\sum_{c=1}^{C} \alpha_c + N}
\]

\[
\hat{\pi}_c = p(y = c) = \frac{N_c}{N}
\]

\[
\bar{\theta}_{jc} = \frac{\beta_1 + N_{jc}}{\beta_0 + \beta_1 + N_c}
\]

\[
\hat{\theta}_{jc} = p(x_j = 1 \mid y = c) = \frac{N_{jc}}{N_c}
\]

\[
= \frac{\text{Number of samples belonging to class } c \text{ that contain feature } j}{\text{Number of samples belonging to class } c}
\]
Bayesian Naïve Bayes Classifier

- Often we just take $\alpha = 1$ and $\beta = 1$.
- This corresponding to **add-one or Laplace** smoothing.

\[
\bar{\pi}_c = \frac{\alpha_c + N_c}{\sum_{c=1}^{C} \alpha_c + N}
\]

\[
\hat{\pi}_c = p(y = c) = \frac{N_c}{N} = \frac{\text{Number of samples belonging to class } c}{\text{Total number of samples}}
\]

\[
\bar{\theta}_{jc} = \frac{\beta_1 + N_{jc}}{\beta_0 + \beta_1 + N_c}
\]

\[
\hat{\theta}_{jc} = p(x_j = 1 \mid y = c) = \frac{N_{jc}}{N_c} = \frac{\text{Number of samples belonging to class } c \text{ that contain feature } j}{\text{Number of samples belonging to class } c}
\]
Multivariate Bernoulli NB: Example

• But how did we get these **new formulas** for frequency count?

By applying the **Bayesian Learning approach**.

\[
\bar{\pi}_c = \frac{\alpha_c + N_c}{\sum_{c=1}^{C} \alpha_c + N} \\
\bar{\theta}_{jc} = \frac{\beta_1 + N_{jc}}{\beta_0 + \beta_1 + N_c} \\
\bar{\theta}_{jc} = p(x_j = 1 | y = c) = \frac{N_{jc}}{N_c}
\]

\[
\hat{\pi}_c = p(y = c) = \frac{N_c}{N} \\
\quad = \frac{\text{Number of samples belonging to class } c}{\text{Total number of samples}}
\]

\[
\hat{\theta}_{jc} = p(x_j = 1 | y = c) = \frac{N_{jc}}{N_c} \\
\quad = \frac{\text{Number of samples belonging to class } c \text{ that contain feature } j}{\text{Number of samples belonging to class } c}
\]