10-701 Introduction to Machine Learning

Naïve Bayes

Readings:
Mitchell Ch. 6.1 – 6.10
Murphy Ch. 3

Matt Gormley
Lecture 3
September 14, 2016
Reminders

• Homework 1:
  – due 9/26/16

• Project Proposal:
  – due 10/3/16
  – start early!
Outline

• **Background:**
  – Maximum likelihood estimation (MLE)
  – Maximum a posteriori (MAP) estimation
  – Example: Exponential distribution

• **Generative Models**

• **Model 0:** Not-so-naïve Model

• **Naïve Bayes**
  – Naïve Bayes Assumption
  – Model 1: Bernoulli Naïve Bayes
  – Model 2: Multinomial Naïve Bayes
  – Model 3: Gaussian Naïve Bayes
  – Model 4: Multiclass Naïve Bayes

• **Smoothing**
  – Add-1 Smoothing
  – Add-λ Smoothing
  – MAP Estimation (Beta Prior)
MLE vs. MAP

Suppose we have data \( \mathcal{D} = \{ x^{(i)} \}_{i=1}^{N} \)

\[
\theta^{\text{MLE}} = \arg\max_{\theta} \prod_{i=1}^{N} p(x^{(i)} | \theta)
\]

Maximum Likelihood Estimate (MLE)
Example: MLE of Exponential Distribution

- pdf of Exponential(\(\lambda\)): \(f(x) = \lambda e^{-\lambda x}\)
- Suppose \(X_i \sim \text{Exponential}(\lambda)\) for \(1 \leq i \leq N\).
- Find MLE for data \(D = \{x^{(i)}\}_{i=1}^{N}\)
- First write down log-likelihood of sample.
- Compute first derivative, set to zero, solve for \(\lambda\).
- Compute second derivative and check that it is concave down at \(\lambda^{\text{MLE}}\).
Example: MLE of Exponential Distribution

- First write down log-likelihood of sample.

\[ \ell(\lambda) = \sum_{i=1}^{N} \log f(x^{(i)}) \]  
\[ = \sum_{i=1}^{N} \log(\lambda \exp(-\lambda x^{(i)})) \]  
\[ = \sum_{i=1}^{N} \log(\lambda) + -\lambda x^{(i)} \]  
\[ = N \log(\lambda) - \lambda \sum_{i=1}^{N} x^{(i)} \]
Example: MLE of Exponential Distribution

- Compute first derivative, set to zero, solve for $\lambda$.

$$
\frac{d\ell(\lambda)}{d\lambda} = \frac{d}{d\lambda} N \log(\lambda) - \lambda \sum_{i=1}^{N} x^{(i)} \quad (1)
$$

$$
= \frac{N}{\lambda} - \sum_{i=1}^{N} x^{(i)} = 0 \quad (2)
$$

$$
\Rightarrow \lambda^{\text{MLE}} = \frac{N}{\sum_{i=1}^{N} x^{(i)}} \quad (3)
$$
Background: MLE

Example: MLE of Exponential Distribution

- pdf of Exponential($\lambda$): \( f(x) = \lambda e^{-\lambda x} \)
- Suppose \( X_i \sim \text{Exponential}(\lambda) \) for \( 1 \leq i \leq N \).
- Find MLE for data \( \mathcal{D} = \{x^{(i)}\}_{i=1}^{N} \)
- First write down log-likelihood of sample.
- Compute first derivative, set to zero, solve for \( \lambda \).
- Compute second derivative and check that it is concave down at \( \lambda^{\text{MLE}} \).
Suppose we have data \( \mathcal{D} = \{ x^{(i)} \}_{i=1}^{N} \)

\[ \theta^{\text{MLE}} = \arg \max_{\theta} \prod_{i=1}^{N} p(x^{(i)} | \theta) \]

\[ \theta^{\text{MAP}} = \arg \max_{\theta} \prod_{i=1}^{N} p(x^{(i)} | \theta) p(\theta) \]
Generative Models

• Specify a **generative story** for how the data was created (e.g. roll a weighted die)

• Given **parameters** (e.g. weights for each side) for the model, we can generate **new data** (e.g. roll the die again)

• Typical **learning** approach is MLE (e.g. find the most likely weights given the data)
Features

• Suppose we want to represent a document ($M$ words) as a vector (vocabulary size $V$)

• How should we do it?

Option 1: Integer vector (word IDs)

$$x = [x_1, x_2, \ldots, x_M] \text{ where } x_m \in \{1, \ldots, K\} \text{ a word id.}$$

Option 2: Binary vector (word indicators)

$$x = [x_1, x_2, \ldots, x_K] \text{ where } x_k \in \{0, 1\} \text{ is a boolean.}$$

Option 3: Integer vector (word counts)

$$x = [x_1, x_2, \ldots, x_K] \text{ where } x_k \in \mathbb{Z}^+ \text{ is a positive integer.}$$
Today’s Goal

To define a generative model of emails of two different classes

(e.g. spam vs. not spam)
**Spam News**

**The Economist**

La paralización

Spain may be heading for its third election in a year

Stubborn Socialists are blocking Mariano Rajoy from forming a centre-right government

Sep 5th 2016 | MADRID | Europe

**The Onion**

ELECTION 2016

Tim Kaine Found Riding Conveyor Belt During Factory Campaign Stop

NEWS IN BRIEF

August 23, 2016

VOL. 52 ISSUE 33
Politics - Politicians - Election 2016 - Tim Kaine

Aiken, SC—Noting that he disappeared for over an hour during a campaign stop meet-and-greet with workers at a Bridgestone tire manufacturing plant, sources confirmed Tuesday that Democratic vice presidential candidate Tim Kaine was finally discovered riding on one of the factory's conveyor belts. "Shortly after we arrived, Tim managed to get out of our sight, but after an extensive search of the facilities, one of our interns found him moving down the assembly line between several radial tires," said senior campaign advisor Mike Henry, adding that Kaine could be seen smiling and laughing as he expertly handled the task, much to the delight of the workers.
Model 0: Not-so-naïve Model?

Generative Story:
1. Flip a weighted coin ($Y$)
2. If heads, roll the red many sided die to sample a document vector ($X$) from the Spam distribution
3. If tails, roll the blue many sided die to sample a document vector ($X$) from the Not-Spam distribution

$$P(X_1, \ldots, X_K, Y) = P(X_1, \ldots, X_K | Y) P(Y)$$

This model is computationally naïve!
Model 0: Not-so-naïve Model?

Generative Story:
1. Flip a weighted coin \((Y)\)
2. If heads, sample a document ID \((X)\) from the Spam distribution
3. If tails, sample a document ID \((X)\) from the Not-Spam distribution

\[
P(X, Y) = P(X|Y)P(Y)
\]

This model is computationally naïve!
Model 0: Not-so-naïve Model?

Flip weighted coin

If HEADS, roll red die

If TAILS, roll blue die

Each side of the die is labeled with a document vector (e.g. [1,0,1,...,1])

<table>
<thead>
<tr>
<th>y</th>
<th>x_1</th>
<th>x_2</th>
<th>x_3</th>
<th>...</th>
<th>x_K</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>0</td>
</tr>
</tbody>
</table>
Outline

• **Background:**
  – Maximum likelihood estimation (MLE)
  – Maximum a posteriori (MAP) estimation
  – Example: Exponential distribution

• **Generative Models**

• **Model 0:** Not-so-naïve Model

• **Naïve Bayes**
  – Naïve Bayes Assumption
  – Model 1: Bernoulli Naïve Bayes
  – Model 2: Multinomial Naïve Bayes
  – Model 3: Gaussian Naïve Bayes
  – Model 4: Multiclass Naïve Bayes

• **Smoothing**
  – Add-1 Smoothing
  – Add-λ Smoothing
  – MAP Estimation (Beta Prior)
Naïve Bayes Assumption

Conditional independence of features:

\[ P(X_1, \ldots, X_K, Y) = P(X_1, \ldots, X_K | Y) P(Y) \]

\[ = \left( \prod_{k=1}^{K} P(X_k | Y) \right) P(Y) \]
Estimating a joint from conditional probabilities

\[
P(A, B | C) = P(A | C) \times P(B | C)
\]

∀ \(a, bc : P(A = a \land B = b | C = c) = P(A = a | C = c) \times P(B = b | C = c)\)
Estimating a joint from conditional probabilities

<table>
<thead>
<tr>
<th></th>
<th>P(C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.33</td>
</tr>
<tr>
<td>1</td>
<td>0.67</td>
</tr>
</tbody>
</table>

| A | C | P(A|C)  |
|---|---|--------|
| 0 | 0 | 0.2    |
| 0 | 1 | 0.5    |
| 1 | 0 | 0.8    |

| B | C | P(B|C)  |
|---|---|--------|
| 0 | 0 | 0.1    |
| 0 | 1 | 0.9    |
| 1 | 0 | 0.9    |
| 1 | 1 | 0.1    |

| D | C | P(D|C)  |
|---|---|--------|
| 0 | 0 | 0.1    |
| 0 | 1 | 0.1    |
| 1 | 0 | 0.9    |
| 1 | 1 | 0.1    |

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>D</th>
<th>C</th>
<th>P(A,B,D,C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Assuming conditional independence, the conditional probabilities encode the **same information** as the joint table.

They are very convenient for estimating

\[
P(X_1,\ldots,X_n|Y) = P(X_1|Y) \ast \ldots \ast P(X_n|Y)
\]

They are almost as good for computing

\[
P(Y|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y)P(Y)}{P(X_1,\ldots,X_n)}
\]

\[
\forall x, y : P(Y = y |X_1,\ldots,X_n = x) = \frac{P(X_1,\ldots,X_n = x |Y)P(Y = y)}{P(X_1,\ldots,X_n = x)}
\]
**Generic Naïve Bayes Model**

**Support:** Depends on the choice of **event model**, $P(X_k|Y)$

**Model:** Product of **prior** and the event model

\[
P(X, Y) = P(Y) \prod_{k=1}^{K} P(X_k|Y)
\]

**Training:** Find the **class-conditional** MLE parameters

For $P(Y)$, we find the MLE using all the data. For each $P(X_k|Y)$ we condition on the data with the corresponding class.

**Classification:** Find the class that maximizes the posterior

\[
\hat{y} = \arg\max_y p(y|x)
\]
Naïve Bayes Model

Classification:

\[ \hat{y} = \arg\max_y p(y|x) \quad \text{(posterior)} \]

\[ = \arg\max_y \frac{p(x|y)p(y)}{p(x)} \quad \text{(by Bayes' rule)} \]

\[ = \arg\max_y p(x|y)p(y) \]
Model 1: Bernoulli Naïve Bayes

**Support:** Binary vectors of length \( K \)
\[
\mathbf{x} \in \{0, 1\}^K
\]

**Generative Story:**
\[
Y \sim \text{Bernoulli}(\phi)
\]
\[
X_k \sim \text{Bernoulli}(\theta_k, Y) \; \forall k \in \{1, \ldots, K\}
\]

**Model:**
\[
p_{\phi, \theta}(\mathbf{x}, y) = p_{\phi, \theta}(x_1, \ldots, x_K, y)
\]
\[
= p_{\phi}(y) \prod_{k=1}^{K} p_{\theta_k}(x_k | y)
\]
\[
= (\phi)^y (1 - \phi)^{1-y} \prod_{k=1}^{K} (\theta_k, y)^{x_k} (1 - \theta_k, y)^{1-x_k}
\]
Model 0: Not-so-naïve Model?

Flip weighted coin

If HEADS, flip each red coin

<table>
<thead>
<tr>
<th>y</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>...</th>
<th>$x_K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>0</td>
</tr>
</tbody>
</table>

If TAILS, flip each blue coin

Each red coin corresponds to an $x_k$

We can generate data in this fashion. Though in practice we never would since our data is given.

Instead, this provides an explanation of how the data was generated (albeit a terrible one).
Model 1: Bernoulli Naïve Bayes

**Support:** Binary vectors of length K

\[ x \in \{0, 1\}^K \]

**Generative Story:**

\[ Y \sim \text{Bernoulli}(\phi) \]

\[ X_k \sim \text{Bernoulli}(\theta_{k,Y}) \quad \forall k \in \{1, \ldots, K\} \]

**Model:**

\[ p_{\phi, \theta}(x, y) = (\phi)^y (1 - \phi)^{(1-y)} \]

**Classification:** Find the class that maximizes the posterior

\[ \hat{y} = \arg\max_y p(y|x) \]

Same as Generic Naïve Bayes
Classification:
\[ \hat{y} = \operatorname{argmax}_{y} p(y|x) \quad \text{(posterior)} \]
\[ = \operatorname{argmax}_{y} \frac{p(x|y)p(y)}{p(x)} \quad \text{(by Bayes’ rule)} \]
\[ = \operatorname{argmax}_{y} p(x|y)p(y) \]
Model 1: Bernoulli Naïve Bayes

**Training:** Find the class-conditional MLE parameters

For $P(Y)$, we find the MLE using all the data. For each $P(X_k | Y)$ we condition on the data with the corresponding class.

$$
\phi = \frac{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1)}{N}
$$

$$
\theta_{k,0} = \frac{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \land x_k^{(i)} = 1)}{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0)}
$$

$$
\theta_{k,1} = \frac{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1 \land x_k^{(i)} = 1)}{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1)}
$$

$\forall k \in \{1, \ldots, K\}$
Model 1: Bernoulli Naïve Bayes

Training: Find the class-conditional MLE parameters

For $P(Y)$, we find the MLE using all the data. For each $P(X_k|Y)$ we condition on the data with the corresponding class.

$$
\phi = \frac{\sum_{i=1}^{N} I(y(i) = 1)}{N}
$$

$$
\theta_{k,0} = \frac{\sum_{i=1}^{N} I(y(i) = 0 \land x_k^{(i)} = 1)}{\sum_{i=1}^{N} I(y(i) = 0)}
$$

$$
\theta_{k,1} = \frac{\sum_{i=1}^{N} I(y(i) = 1 \land x_k^{(i)} = 1)}{\sum_{i=1}^{N} I(y(i) = 1)}
$$

$\forall k \in \{1, \ldots, K\}$
Outline

• **Background:**
  – Maximum likelihood estimation (MLE)
  – Maximum a posteriori (MAP) estimation
  – Example: Exponential distribution

• **Generative Models**

• **Model 0:** Not-so-naïve Model

• **Naïve Bayes**
  – Naïve Bayes Assumption
  – Model 1: Bernoulli Naïve Bayes
  – Model 2: Multinomial Naïve Bayes
  – Model 3: Gaussian Naïve Bayes
  – Model 4: Multiclass Naïve Bayes

• **Smoothing**
  – Add-1 Smoothing
  – Add-λ Smoothing
  – MAP Estimation (Beta Prior)
Smoothing

1. Add-1 Smoothing
2. Add-λ Smoothing
3. MAP Estimation (Beta Prior)
MLE

What does maximizing likelihood accomplish?

• There is only a finite amount of probability mass (i.e. sum-to-one constraint)

• MLE tries to allocate as much probability mass as possible to the things we have observed...

...at the expense of the things we have not observed
MLE

For Naïve Bayes, suppose we never observe the word “serious” in an Onion article.
In this case, what is the MLE of \( p(x_k \mid y) \)?

\[
\theta_{k,0} = \frac{\sum_{i=1}^{N} \mathbb{I}(y(i) = 0 \land x_{k}^{(i)} = 1)}{\sum_{i=1}^{N} \mathbb{I}(y(i) = 0)}
\]

Now suppose we observe the word “serious” at test time. What is the posterior probability that the article was an Onion article?

\[
p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)}
\]
1. Add-1 Smoothing

The simplest setting for smoothing simply adds a single pseudo-observation to the data. This converts the true observations $\mathcal{D}$ into a new dataset $\mathcal{D}'$ from which we derive the MLEs.

$$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$$  \hspace{1cm} (1)

$$\mathcal{D}' = \mathcal{D} \cup \{(0, 0), (0, 1), (1, 0), (1, 1)\}$$  \hspace{1cm} (2)

where $\mathbf{0}$ is the vector of all zeros and $\mathbf{1}$ is the vector of all ones.

This has the effect of pretending that we observed each feature $x_k$ with each class $y$. 
1. Add-1 Smoothing

What if we write the MLEs in terms of the original dataset \( \mathcal{D} \)?

\[ \phi = \frac{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1)}{N} \]

\[ \theta_{k,0} = \frac{1 + \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \land x^{(i)}_k = 1)}{2 + \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0)} \]

\[ \theta_{k,1} = \frac{1 + \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1 \land x^{(i)}_k = 1)}{2 + \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1)} \]

\[ \forall k \in \{1, \ldots, K\} \]
2. Add-\(\lambda\) Smoothing

For the Categorical Distribution

Suppose we have a dataset obtained by repeatedly rolling a \(K\)-sided (weighted) die. Given data \(\mathcal{D} = \{x^{(i)}\}_{i=1}^N\) where \(x^{(i)} \in \{1, \ldots, K\}\), we have the following MLE:

\[
\phi_k = \frac{\sum_{i=1}^N \mathbb{I}(x^{(i)} = k)}{N}
\]

With add-\(\lambda\) smoothing, we add pseudo-observations as before to obtain a smoothed estimate:

\[
\phi_k = \frac{\lambda + \sum_{i=1}^N \mathbb{I}(x^{(i)} = k)}{k\lambda + N}
\]
MLE vs. MAP

Suppose we have data \( \mathcal{D} = \{ x^{(i)} \}_{i=1}^{N} \)

Maximum Likelihood Estimate (MLE)

\[
\theta^{\text{MLE}} = \arg\max_\theta \prod_{i=1}^{N} p(x^{(i)} | \theta)
\]

Maximum a posteriori (MAP) estimate

\[
\theta^{\text{MAP}} = \arg\max_\theta \prod_{i=1}^{N} p(x^{(i)} | \theta) p(\theta)
\]

Prior
3. MAP Estimation (Beta Prior)

Generative Story:
The parameters are drawn once for the entire dataset.

\[
\begin{align*}
\text{for } k \in \{1, \ldots, K\}: \\
\quad & \text{for } y \in \{0, 1\}: \\
\quad \quad & \theta_{k,y} \sim \text{Beta}(\alpha, \beta) \\
\text{for } i \in \{1, \ldots, N\}: \\
\quad & y^{(i)} \sim \text{Bernoulli}(\phi) \\
& \text{for } k \in \{1, \ldots, K\}: \\
& \quad x^{(i)}_k \sim \text{Bernoulli}(\theta_{k,y^{(i)}})
\end{align*}
\]

Training: Find the class-conditional MAP parameters

\[
\phi = \frac{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1)}{N}
\]

\[
\begin{align*}
\theta_{k,0} &= \frac{(\alpha - 1) + \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \land x^{(i)}_k = 1)}{(\alpha - 1) + (\beta - 1) + \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0)} \\
\theta_{k,1} &= \frac{(\alpha - 1) + \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1 \land x^{(i)}_k = 1)}{(\alpha - 1) + (\beta - 1) + \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1)}
\end{align*}
\]

\[\forall k \in \{1, \ldots, K\}\]
Outline

• **Background:**
  – Maximum likelihood estimation (MLE)
  – Maximum a posteriori (MAP) estimation
  – Example: Exponential distribution

• **Generative Models**

• **Model 0:** Not-so-naïve Model

• **Naïve Bayes**
  – Naïve Bayes Assumption
  – Model 1: Bernoulli Naïve Bayes
  – Model 2: Multinomial Naïve Bayes
  – Model 3: Gaussian Naïve Bayes
  – Model 4: Multiclass Naïve Bayes

• **Smoothing**
  – Add-1 Smoothing
  – Add-λ Smoothing
  – MAP Estimation (Beta Prior)
Model 2: Multinomial Naïve Bayes

**Support:**

Option 1: Integer vector (word IDs)

\[
x = [x_1, x_2, \ldots, x_M] \text{ where } x_m \in \{1, \ldots, K\} \text{ a word id.}
\]

**Generative Story:**

\[
\text{for } i \in \{1, \ldots, N\}:
\]
\[
y^{(i)} \sim \text{Bernoulli}(\phi)
\]

\[
\text{for } j \in \{1, \ldots, M_i\}:
\]
\[
x_j^{(i)} \sim \text{Multinomial}(\theta_{y^{(i)}}, 1)
\]

**Model:**

\[
p_{\phi, \theta}(x, y) = p_{\phi}(y) \prod_{k=1}^{K} p_{\theta_k}(x_k | y)
\]

\[
= (\phi)^y (1 - \phi)^{(1-y)} \prod_{j=1}^{M_i} \theta_{y, x_j}
\]
### Gaussian Discriminative Analysis

- **learning f: X → Y, where**
  - X is a vector of real-valued features, \( X_n = < X_{n1}, \ldots, X_{nm} > \)
  - Y is an indicator vector

- **What does that imply about the form of \( P(Y|X) \)?**
  - The joint probability of a datum and its label is:
    \[
    p(x_n, y_n^k = 1 \mid \mu, \sigma) = p(y_n^k = 1) \times p(x_n \mid y_n^k = 1, \mu, \Sigma) \\
    = \pi_k \frac{1}{(2\pi |\Sigma|)^{1/2}} \exp\left\{ \frac{1}{2} (x_n - \bar{\mu}_k)^T \Sigma^{-1} (x_n - \bar{\mu}_k) \right\}
    \]

  - Given a datum \( x_n \), we predict its label using the conditional probability of the label given the datum:
    \[
    p(y_n^k = 1 \mid x_n, \mu, \Sigma) = \frac{\pi_k \frac{1}{(2\pi |\Sigma|)^{1/2}} \exp\left\{ \frac{1}{2} (x_n - \bar{\mu}_k)^T \Sigma^{-1} (x_n - \bar{\mu}_k) \right\}}{\sum_k \pi_k \frac{1}{(2\pi |\Sigma|)^{1/2}} \exp\left\{ \frac{1}{2} (x_n - \bar{\mu}_k)^T \Sigma^{-1} (x_n - \bar{\mu}_k) \right\}}
    \]
Model 3: Gaussian Naïve Bayes

Support: \( \mathbf{x} \in \mathbb{R}^K \)

**Model:** Product of prior and the event model

\[
p(x, y) = p(x_1, \ldots, x_K, y) = p(y) \prod_{k=1}^{K} p(x_k | y)
\]

Gaussian Naïve Bayes assumes that \( p(x_k | y) \) is given by a Normal distribution.
Gaussian Naïve Bayes Classifier

- When X is multivariate-Gaussian vector:
  - The joint probability of a datum and it label is:

\[
p(x_n, y_n^k = 1 \mid \bar{\mu}, \bar{\Sigma}) = p(y_n^k = 1) \times p(x_n \mid y_n^k = 1, \bar{\mu}, \bar{\Sigma})
\]

\[
= \pi_k \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp \left\{ -\frac{1}{2} (x_n - \bar{\mu}_k)^T \Sigma^{-1} (x_n - \bar{\mu}_k) \right\}
\]

- The naïve Bayes simplification

\[
p(x_n, y_n^k = 1 \mid \mu, \sigma) = p(y_n^k = 1) \times \prod_j p(x_n^j \mid y_n^k = 1, \mu_k^j, \sigma_k^j)
\]

\[
= \pi_k \prod_j \frac{1}{\sqrt{2\pi\sigma_k^j}} \exp \left\{ -\frac{1}{2} \left( \frac{x_n^j - \mu_k^j}{\sigma_k^j} \right)^2 \right\}
\]

- More generally:

  - Where \( p(., \mid .) \) is an arbitrary conditional (discrete or continuous), 1-D density

\[
p(x_n, y_n \mid \eta, \bar{x}) = p(y_n \mid \bar{x}) \times \prod_j p(x_n^j \mid y_n, \eta)
\]
VISUALIZING NAÏVE BAYES
Fisher Iris Dataset

Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

<table>
<thead>
<tr>
<th>Species</th>
<th>Sepal Length</th>
<th>Sepal Width</th>
<th>Petal Length</th>
<th>Petal Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4.3</td>
<td>3.0</td>
<td>1.1</td>
<td>0.1</td>
</tr>
<tr>
<td>0</td>
<td>4.9</td>
<td>3.6</td>
<td>1.4</td>
<td>0.1</td>
</tr>
<tr>
<td>0</td>
<td>5.3</td>
<td>3.7</td>
<td>1.5</td>
<td>0.2</td>
</tr>
<tr>
<td>1</td>
<td>4.9</td>
<td>2.4</td>
<td>3.3</td>
<td>1.0</td>
</tr>
<tr>
<td>1</td>
<td>5.7</td>
<td>2.8</td>
<td>4.1</td>
<td>1.3</td>
</tr>
<tr>
<td>1</td>
<td>6.3</td>
<td>3.3</td>
<td>4.7</td>
<td>1.6</td>
</tr>
<tr>
<td>1</td>
<td>6.7</td>
<td>3.0</td>
<td>5.0</td>
<td>1.7</td>
</tr>
</tbody>
</table>

Full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set
%%% Import the IRIS data
load fisheriris;
X = meas;
pos = strcmp(species,'setosa');
Y = 2*pos - 1;

%%% Visualize the data
imagesc([X,Y]);
title('Iris data');
%% Visualize by scatter plotting the first two dimensions

figure;
scatter(X(Y<0,1),X(Y<0,2),'r*');
hold on;
scatter(X(Y>0,1),X(Y>0,2),'bo');
title('Iris data');
%% Compute the mean and SD of each class
PosMean = mean(X(Y>0,:));
PosSD = std(X(Y>0,:));
NegMean = mean(X(Y<0,:));
NegSD = std(X(Y<0,:));

%% Compute the NB probabilities for each class for each grid element
[G1,G2]=meshgrid(3:0.1:8, 2:0.1:5);
Z1 = gaussmf(G1,[PosSD(1),PosMean(1)]);
Z2 = gaussmf(G2,[PosSD(2),PosMean(2)]);
Z = Z1.* Z2;

V1 = gaussmf(G1,[NegSD(1),NegMean(1)]);
V2 = gaussmf(G2,[NegSD(2),NegMean(2)]);
V = V1.* V2;
% Add them to the scatter plot
figure;
saveas(gcf, 'myplot.png');
hold on;
saveas(gcf, 'myplot.png');
scatter(X(Y<0,1),X(Y<0,2),'r*');
hold on;
scatter(X(Y>0,1),X(Y>0,2),'bo');
contour(G1,G2,Z); 
contour(G1,G2,V);
%% Now plot the difference of the probabilities
figure;
scatter(X(Y<0,1),X(Y<0,2),'r*');
hold on;
scatter(X(Y>0,1),X(Y>0,2),'bo');
contour(G1,G2,Z-V);
mesh(G1,G2,Z-V);
alpha(0.4)
NAÏVE BAYES IS LINEAR
Question: what does the *boundary* between positive and negative look like for Naïve Bayes?
argmax_y \prod_i P(X_i = x_i | Y = y) P(Y = y)

= \argmax_y \sum_i \log P(X_i = x_i | Y = y) + \log P(Y = y)

= \argmax_y \sum_{y \in \{+1, -1\}} \log P(x_i | y) + \log P(y)

= \text{sign} \left( \sum_i \log P(x_i | y_+) - \sum_i \log P(x_i | y_-) + \log P(y_+) - \log P(y_-) \right)

= \text{sign} \left( \sum_i \log \frac{P(x_i | y_+)}{P(x_i | y_-)} + \log \frac{P(y_+)}{P(y_-)} \right)

two classes only

rearrange terms
\[
\operatorname{argmax}_y \prod_i P(X_i = x_i \mid Y = y) P(Y = y)
\]

\[
= \text{sign}\left( \sum_i \log \frac{P(x_i \mid y_+)}{P(x_i \mid y_-)} + \log \frac{P(y_+)}{P(y_-)} \right)
\]

if \(x_i = 1\) or 0 ....

\[
u_i = \left( \log \frac{P(x_i = 1 \mid y_+)}{P(x_i = 1 \mid y_-)} - \log \frac{P(x_i = 0 \mid y_+)}{P(x_i = 0 \mid y_-)} \right)
\]

\[
= \text{sign}\left( \sum_i x_i u_i + u_0 \right)
\]

\[
u_0 = \sum_i \left( \log \frac{P(x_i = 0 \mid y_+)}{P(x_i = 0 \mid y_-)} + \log \frac{P(y_+)}{P(y_-)} \right)
\]

\[
x_0 = 1 \text{ for every } \mathbf{x} \text{ (bias term)}
\]

\[= \text{sign}(\mathbf{x} \cdot \mathbf{u})\]
Why don’t we drop the generative model and try to learn this hyperplane directly?
Model 4: Multiclass Naïve Bayes

Model:

The only change is that we permit $y$ to range over $C$ classes.

$$p(x, y) = p(x_1, \ldots, x_K, y)$$

$$= p(y) \prod_{k=1}^{K} p(x_k | y)$$

Now, $y \sim \text{Multinomial}(\phi, 1)$ and we have a separate conditional distribution $p(x_k | y)$ for each of the $C$ classes.
Naïve Bayes Assumption

Conditional independence of features:

\[ P(X_1, \ldots, X_K, Y) = P(X_1, \ldots, X_K | Y)P(Y) \]

\[ = \left( \prod_{k=1}^{K} P(X_k | Y) \right) P(Y) \]
What’s wrong with the Naïve Bayes Assumption?

The features might not be independent!!

• Example 1:
  – If a document contains the word “Donald”, it’s extremely likely to contain the word “Trump”
  – These are not independent!

• Example 2:
  – If the petal width is very high, the petal length is also likely to be very high
Summary

1. Naïve Bayes provides a framework for **generative modeling**
2. Choose an **event model** appropriate to the data
3. Train by **MLE** or **MAP**
4. Classify by maximizing the posterior