Machine Learning

Generative Classification & Naïve Bayes

Dariush Hosseini

dariush.hosseini@ucl.ac.uk
Department of Computer Science
University College London

Lecture Overview

Lecture Overview

By the end of this lecture you should:

1. Understand the **Naïve Bayes** algorithm and its motivation as a **Generative** approach to the classification problem
2. Understand **discrete** and **continuous** version of the Naïve Bayes algorithm
3. Understand the relationship between **Gaussian Naïve Bayes** and **Logistic Regression**

Lecture Overview

Notation

- **Inputs**
  $\mathbf{x} = [1, x_1, ..., x_m]^T \in \mathbb{R}^{m+1}$
- **Binary Outputs**
  $y \in \{0, 1\}$
- **Training Data**
  $\mathcal{S} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$
- **Data-Generating Distribution**, $\mathcal{D}$
  $\mathcal{S} \sim \mathcal{D}$

Probabilistic Environment

- We assume:
  - $\mathbf{x}$ is the outcome of a random variable $\mathcal{X}$
  - y is the outcome of a random variable $\mathcal{Y}$
  - $(\mathbf{x}, y)$ are drawn i.i.d. from some data generating distribution, $\mathcal{D}$, i.e.:

$$(\mathbf{x}, y) \sim \mathcal{D}$$

  and:

$$\mathcal{S} \sim \mathcal{D}^n$$

Learning Problem

- **Representation**

$$f \in \mathcal{F}$$

- **Evaluation**
  - **Loss Measure:**

$$\mathcal{E}(f(\mathbf{x}), y) = \mathbb{I}[y \neq f(\mathbf{x})]$$

  - **Generalisation Loss:**

$$\mathsf{L}(\mathcal{E}, \mathcal{D}, f) = \mathbb{E}_{\mathcal{D}}\left[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]\right]$$

Where $\mathcal{D}$ is characterised by $p_{\mathcal{X},\mathcal{Y}}(\mathbf{x}, y) = p_{\mathcal{Y}}(y|\mathbf{x})p_{\mathcal{X}}(\mathbf{x})$ for some pmf, $p_{\mathcal{Y}}(\cdot|\cdot)$, and some pdf, $p_{\mathcal{X}}(\cdot)$

- **Optimisation**

$$f^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}}\, \mathbb{E}_{\mathcal{D}}\left[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]\right]$$

Bayes Optimal Classifier

- So the generalisation minimiser for the **Misclassification Loss** can be specified entirely in term of the **posterior distribution**:

$$f^*(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad p_{\mathcal{Y}}(y = 1|\mathbf{x}) \geqslant 0.5 \\ 0 & \text{if} \quad p_{\mathcal{Y}}(y = 1|\mathbf{x}) < 0.5 \end{cases}$$

- It is known as the **Bayes Optimal Classifier**

Probabilistic Classifier

- In probabilistic classification we use this expression for the Bayes Optimal Classifier in order to re-cast the classification problem as an **inference problem** in which we must learn $p_{\mathcal{Y}}(y = 1|\mathbf{x})$
- Here $p_{\mathcal{Y}}(y = 1|\mathbf{x})$ characterises an **inhomogeneous Bernoulli distribution**

Generative Classification

- In **Generative Classification** we seek to learn $p_{\mathcal{Y}}(y = 1|\mathbf{x})$ **indirectly**

- First we re-express the Bayes Optimal Classifier as follows, without loss of generality:

$$
\begin{aligned}
f^*(\mathbf{x}) &= \underset{y \in \{0,1\}}{\text{argmax}} \quad p_{\mathcal{Y}}(y|\mathbf{x}) \\
&= \underset{y \in \{0,1\}}{\text{argmax}} \quad \frac{p_{\mathcal{X}}(\mathbf{x}|y) p_{\mathcal{Y}}(y)}{\sum_{y \in \{0,1\}} p_{\mathcal{X}}(\mathbf{x}|y) p_{\mathcal{Y}}(y)} && \textbf{Bayes' Theorem} \\
&= \underset{y \in \{0,1\}}{\text{argmax}} \quad p_{\mathcal{X}}(\mathbf{x}|y) p_{\mathcal{Y}}(y) && \textbf{Denominator doesn't depend on } y
\end{aligned}
$$

- Then we seek to infer the **likelihood** $p_{\mathcal{X}}(\mathbf{x}|y)$ and the **prior** $p_{\mathcal{Y}}(y)$ for each class separately

Inference Problem

- Inferring $p_{\mathcal{Y}}(y)$ is straightforward
  - In binary classification there is only one parameter to learn
- Inferring $p_{\mathcal{X}}(\mathbf{x}|y)$ is more difficult
  - For example: consider $\mathbf{x}$ which is a vector of **boolean** functions
  - For each possible value of $\mathbf{x} = \widehat{\mathbf{x}}$ and $y = \widehat{y}$ we must learn a probability, $p_{\mathcal{X}}(\widehat{\mathbf{x}}|\widehat{y})$
  - For each value of $\widehat{y}$ there are 2^m possible values of $\widehat{\mathbf{x}}$
  - $2^m - 1$ parameters must be inferred for each output class
  - And $2(2^m - 1)$ parameters must be inferred altogether
  - This is **intractable**

Example: Document Topic Classification

- Outcomes, y, of a random variable, $\mathcal{Y}$, characterise a set of **topics**
- Outcomes, $\mathbf{x}$, of a random variable, $\mathcal{X}$, characterise a particular document according to the **bag-of-words** representation
- Here word order doesn't matter, instead $\mathbf{x}$ is a vector whose elements are boolean, each of which indicates the presence or absence of a particular **dictionary** word in the document
- A **dictionary** is the set of words

Example: Document Topic Classification

- So, for example:

$$\mathbf{x}^{(i)} = \begin{bmatrix} \text{`aardvark'} : \mathbf{x}_1^{(i)} = 1 \\ \vdots \\ \text{`zyme'} : \mathbf{x}_m^{(i)} = 0 \end{bmatrix}$$

- But a dictionary contains $\sim 10,000$ words

- So $m \approx 10,000$, and we need to infer $\sim 2^{10,000}$ parameters to characterise the likelihood!

- We need a simplifying assumption...

Lecture Overview

Conditional Independence: Definition

- Given 3 random variables, $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$, we say that $\mathcal{X}$ is **conditionally independent** of $\mathcal{Y}$ given $\mathcal{Z}$ iff the probability distribution governing $\mathcal{X}$ is independent of the outcomes of $\mathcal{Y}$ given the outcomes of $\mathcal{Z}$

So, $\forall i, j, k$:

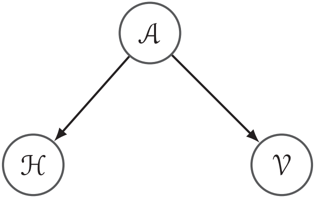
$$\mathbb{P}\left(x^{(i)}|y^{(j)}, z^{(k)}\right) = \mathbb{P}\left(x^{(i)}|z^{(k)}\right)$$

$$\implies \mathbb{P}\left(x^{(i)}|y^{(j)}, z^{(k)}\right)\mathbb{P}\left(y^{(j)}|z^{(k)}\right) = \mathbb{P}\left(x^{(i)}|z^{(k)}\right)\mathbb{P}\left(y^{(j)}|z^{(k)}\right)$$

$$\implies \mathbb{P}\left(x^{(i)}, y^{(j)}|z^{(k)}\right) = \mathbb{P}\left(x^{(i)}|z^{(k)}\right)\mathbb{P}\left(y^{(j)}|z^{(k)}\right)$$

Here $x^{(i)}, y^{(j)}, z^{(k)}$ are outcomes of $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ respectively
And the notation $\mathbb{P}\left(x^{(i)}|y^{(j)}, z^{(k)}\right)$ is used as a short-hand for $\mathbb{P}\left(\mathcal{X} = x^{(i)}|\mathcal{Y} = y^{(j)}, \mathcal{Z} = z^{(k)}\right)$

Conditional Independence: Example



- $\mathcal{A}$ is a random variable with outcomes that are children's ages
- $\mathcal{H}$ is a random variable with outcomes that are children's heights
- $\mathcal{V}$ is a random variable with outcomes that are the ranges of children's vocabulary

- $\mathbb{P}(\mathcal{H} = h, \mathcal{V} = v) \neq \mathbb{P}(\mathcal{H} = h)\mathbb{P}(\mathcal{V} = v)$
- $\mathbb{P}(\mathcal{H} = h, \mathcal{V} = v|\mathcal{A} = a) = \mathbb{P}(\mathcal{H} = h|\mathcal{A} = a)\mathbb{P}(\mathcal{V} = v|\mathcal{A} = a)$

Naïve Bayes

- Recall that each sample, $(\mathbf{x}, y)$ is an outcome of a random variable, $\mathcal{X}, \mathcal{Y}$
- Furthermore: Each element of $\mathbf{x}$, x_i, is the outcome of a corresponding random variable, $\mathcal{X}_i$
- Thus: $p_{\mathcal{X}}(\mathbf{x}) = p_{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_m}(x_1, x_2, ..., x_m)$
- **Naïve Bayes** seeks to **simplify** the likelihood by assuming that $\{\mathcal{X}_i\}_{i=1}^m$ are all **conditionally independent** given $\mathcal{Y}$:

$$p_{\mathcal{X}}(\mathbf{x}|y) = \prod_{i=1}^{m} p_{\mathcal{X}_i}(x_i|y)$$

- This is a much simpler representation
- So: For our vector of boolean attributes we now need only $2m$ parameters, rather than $2^m - 1$, to characterise the likelihood

Example: Document Topic Classification

- Consider again our **bag-of-words** example
  - Let the **topic** of y be **'Politics'**
  - Let x_i correspond to the presence or absence of the word **'Trump'**
  - Let x_j correspond to the presence or absence of the word **'Clinton'**
- The **conditional independence** assumption implies that:

$$\mathbb{P}(x_i = \textbf{'Trump'} | x_j = \textbf{'Clinton'}, y = \textbf{'Politics'}) = \mathbb{P}(x_i = \textbf{'Trump'} | y = \textbf{'Politics'})$$

- This is quite a strong assumption...surely:

$$\mathbb{P}(x_i = \textbf{'Trump'} | x_j = \textbf{'Clinton'}, y = \textbf{'Politics'}) > \mathbb{P}(x_i = \textbf{'Trump'} | y = \textbf{'Politics'})$$

- Despite this Naïve Bayes often works well as a classifier

Representation

- Recall that we seek:

$$f^*(\mathbf{x}) = \underset{y \in \{0,1\}}{\operatorname{argmax}} \quad p_{\mathcal{X}}(\mathbf{x}|y) p_{\mathcal{Y}}(y)$$

$$= \underset{y \in \{0,1\}}{\operatorname{argmax}} \quad p_{\mathcal{Y}}(y) \prod_{i=1}^{m} p_{\mathcal{X}_i}(x_i|y) \quad \textbf{By NB assumption}$$

- But how do we learn the parameterisation of the prior and the likelihood?
- Let's consider two cases:
  - **Categorical Naïve Bayes** $\longrightarrow$ for **discrete** inputs
  - **Gaussian Naïve Bayes** $\longrightarrow$ for **continuous** inputs

Categorical Naïve Bayes

- Assume that the features, x_i, are discrete-valued, and take m_i different values, such that the outcomes of x_i are taken from the set $\{x_{ij}\}_{j=1}^{m_i}$
- Let us attempt to learn the pmf's for $p_{\mathcal{Y}}(y)$ and $p_{\mathcal{X}_i}(x_i|y)$ in a **frequentist** setting using **MLE**

Evaluation: $p_{\mathcal{Y}}(y)$

- $\mathcal{Y}$ is a Bernoulli random variable, with outcomes, $y \sim \text{Bern}(\theta_y)$, which implies the following log-likelihood function:

$$\begin{aligned}
\ln\left(\text{L}(\theta)\right) &= \ln\left(\prod_{i=1}^{n} p_{\mathcal{Y}}(y^{(i)}; \theta_y)\right) \\
&= \sum_{i=1}^{n} \ln\left(p_{\mathcal{Y}}(y^{(i)}; \theta_y)\right) \\
&= \sum_{i=1}^{n} y^{(i)} \ln \theta_y + (1 - y^{(i)}) \ln\left(1 - \theta_y\right)
\end{aligned}$$

Optimisation: $p_{\mathcal{Y}}(y)$

- We seek $\theta_{y\text{MLE}}$ such that:

$$\theta_{y\text{MLE}} = \underset{\theta_y}{\text{argmax}} \sum_{i=1}^{n} y^{(i)} \ln \theta_y + (1 - y^{(i)}) \ln (1 - \theta_y)$$

- Let's try to find an analytic solution:

$$\frac{d}{d\theta_y} \ln (\mathrm{L}(\theta)) = \sum_{i=1}^{n} \frac{y^{(i)}}{\theta_y} - \frac{(1 - y^{(i)})}{1 - \theta_y}$$

Optimisation: $p_{\mathrm{y}}(y)$

- For stationarity set this equal to zero:

$$\sum_{i=1}^{n} \frac{y^{(i)}}{\theta_{y\text{MLE}}} - \frac{(1-y^{(i)})}{1-\theta_{y\text{MLE}}} = 0$$

$$\implies \quad \sum_{i=1}^{n} y^{(i)}(1-\theta_{y\text{MLE}}) - (1-y^{(i)})\theta_{y\text{MLE}} = 0$$

$$\implies \quad \sum_{i=1}^{n} y^{(i)} = \sum_{i=1}^{n} \theta_{y\text{MLE}}$$

$$\implies \quad \theta_{y\text{MLE}} = \frac{\sum_{i=1}^{n} y^{(i)}}{n} = \frac{n_1}{n}$$

  Where n_1 is equal to the number of training points for which $y = 1$

- We can demonstrate **convexity** by taking the second derivative

Evaluation: $p_{\mathcal{X}_i}(x_i|y)$

- $(\mathcal{X}_i|y = k)$ is a **categorical** random variable, which can take the values $\{x_{ij}\}_{j=1}^{m_i}$
- We seek to parameterise a different categorical distribution for each $(\mathcal{X}_i, y = k)$
- What is the categorical distribution?

Evaluation: $p_{\mathcal{X}_i}(x_i|y)$

- It is a generalisation of the Bernoulli distribution, where the random variable has more than 2 discrete outcomes (in this case m_i):

$$(\mathbf{x}_i|y = k) \sim \text{Categorical}(\Theta_{ik})$$

$$\Theta_{ik} \quad \text{has elements} \quad \{\theta_{ijk}\}_{j=1}^{m_i}$$

$$\sum_{j=1}^{m_i} \theta_{ijk} = 1 \quad \text{has elements} \quad \{\theta_{ijk}\}_{j=1}^{m_i}$$

$$p_{\mathcal{X}_i}(\mathcal{X}_i = x_{ij}|y = k; \Theta_{ik}) = \theta_{ijk}$$

Evaluation & Optimisation: $p_{\mathcal{X}_i}(x_i|y)$

- We can learn these pmf's by forming the log-likelihood, and then performing a constrained optimisation of the resulting function using the method of Lagrange multipliers

- This results in:

$$\theta_{ijk\text{MLE}} = \frac{n_{ijk}}{n_k}$$

Where n_{ijk} is equal to the number of training points for which $(\mathcal{X}_i = x_{ij} \wedge \mathcal{Y} = k)$

Where n_k is equal to the number of training points for which $\mathcal{Y} = k$

UCL

Recap

- **Representation**

$$\mathcal{F} = \left\{ f_{\theta_y, \{\theta_{ijk}\}}(\mathbf{x}) = \underset{y \in \{0,1\}}{\operatorname{argmax}} \quad p_{\mathcal{Y}}(y) \prod_{i=1}^{m} p_{\mathcal{X}_i}(x_i|y) \middle| p_{\mathcal{Y}}(y=1) = \theta_y, \right.$$
$$\left. \left\{ p_{\mathcal{X}_i}(x_{ij}|k) = \theta_{ijk} \right\}_{i=1, j=1, k=0}^{m, m_i, 1} \right\}$$

- **Evaluation**

$$\ln\left(\mathrm{L}(\theta_y)\right) \qquad \text{and} \qquad \left\{ \ln\left(\mathrm{L}(\Theta_{ik})\right) \right\}_{i=1, k=0}^{m,1}$$

- **Optimisation**

$$\theta_{y\mathrm{MLE}} = \frac{n_1}{n} \qquad \text{and} \qquad \left\{ \theta_{ijk\mathrm{MLE}} = \frac{n_{ijk}}{n_k} \right\}_{i=1, j=1, k=0}^{m, m_i, 1}$$

UCL

Problem: Overfitting

- We must take care when the training data contains no instances that satisfy $\mathcal{X}_i = x_{ij}$
- If this occurs then the resulting parameter, θ_{ijk}, would be zero and for any data point for which $\mathcal{X}_i = x_{ij}$, regardless of the state of $\mathcal{Y}$, then:

$$
\begin{aligned}
& p_{\mathcal{X}_i}(x_{ij}|k) = 0 && \forall k \\
\implies & p_{\mathcal{X}}(\mathbf{x}|k) = \prod_{i=1}^{m} p_{\mathcal{X}_i}(x_{ij}|k) = 0 && \forall k \\
\implies & p_{\mathcal{Y}}(k|\mathbf{x}) = \frac{p_{\mathcal{X}}(\mathbf{x}|k)p_{\mathcal{Y}}(k)}{\sum_{\tilde{k}} p_{\mathcal{X}}(\mathbf{x}|\tilde{k})p_{\mathcal{Y}}(\tilde{k})} = \frac{0}{0} && \forall k
\end{aligned}
$$

- And we have a problem!
- This is an example of **overfitting**
  - statistically speaking it's a bad idea to estimate the probability of an event to be zero just because we have never observed it

Solution: Additive Smoothing

- We remedy the problem by adjusting our MLE estimates such that:

$$\theta_{ijk} = \frac{n_{ijk} + \alpha}{n_k + \alpha J}$$
$$\theta_y = \frac{n_1 + \alpha}{n + \alpha K}$$

- Where:

  $J =$ # of distinct values outcomes of $\mathcal{X}_i$ can take (m_i in this case)

  $K =$ # of distinct values outcomes of $\mathcal{Y}$ can take (2 in this case)

  α indicates the strength of **smoothing**

- **Additive smoothing** is like adding instances uniformly to the data

Solution: Additive Smoothing

- Where does additive smoothing come from?
- It is a form of **regularisaton** that emerges most naturally from the **Bayesian approach**:
  - We treat θ_{ijk} and θ_j as random variables and place **prior distributions** over them which correspond to the belief that they are both finite
  - If we choose symmetric **Dirichelet** distributions for each of these priors then the **expectations** of θ_{ijk} and θ_j with respect to their **posterior distributions** yields the additive smoothing estimators
  - Note these are distinct from the MAP estimators

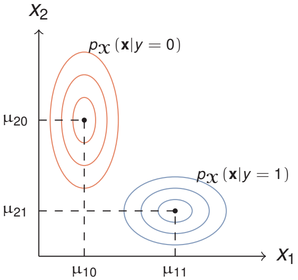
Gaussian Naïve Bayes

- Now let us assume that the features, x_i, are **continuous valued**, such that $x_i \in \mathbb{R}$
- We assume further that $(\mathcal{X}_i|y = k)$ is a **Gaussian** random variable
- Then we attempt to learn the pmf for $p_{\mathcal{Y}}(y)$ and the pdf for $p_{\mathcal{X}_i}(x_i|y)$ in a **frequentist** setting using **MLE**
- The inference problem for $p_{\mathcal{Y}}(y)$ remains the same

Evaluation: $p_{\mathcal{X}_i}(x_i|y)$

$$(x_i|y = k) \sim \mathcal{N}(\mu_{ik}, \sigma_{ik})$$

$$p_{\mathcal{X}_i}(x_i|k; \mu_{ik}, \sigma_{ik}) = \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} e^{-\frac{(x_i - \mu_{ik})^2}{2\sigma_{ik}^2}}$$



Evaluation: $p_{\mathcal{X}_i}(x_i|y)$

- Given $\{x_i^{(j)}\}_{j=1}^{n_k}$ samples drawn from the Normal distribution, and for which $y = k$, the log-likelihood is given by:

$$\ln\left(\mathrm{L}(\mu_{ik}, \sigma_{ik})\right) = \ln\left(\prod_{j=1}^{n_k} \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} e^{-\frac{(x_i^{(j)}-\mu_{ik})^2}{2\sigma_{ik}^2}}\right)$$

$$= -n_k \ln \sigma_{ik} - \sum_{j=1}^{n_k}\left(\frac{(x_i^{(j)} - \mu_{ik})^2}{2\sigma_{ik}^2}\right) + \text{const.}$$

Optimisation: $p_{\mathcal{X}_i}(x_i|y)$

- We can optimise the log-likelihood as we did in the *Statistics* Lecture to give:

$$\mu_{ik\text{MLE}} = \sum_{j=1}^{n_k} \frac{x_i^{(j)}}{n_k}$$

$$\sigma^2_{ik\text{MLE}} = \frac{1}{n_k} \sum_{j=1}^{n_k} \left(x_i^{(j)} - \mu_{ik} \right)^2$$

Recap

- **Representation**

$$\mathcal{F} = \left\{ f_{\theta_y,\{\mu_{ik},\sigma_{ik}\}}(\mathbf{x}) = \underset{y\in\{0,1\}}{\text{argmax}} \quad p_{\mathcal{Y}}(y) \prod_{i=1}^{m} \mathcal{N}(x_i; \mu_{ik}, \sigma_{ik}) \middle| p_{\mathcal{Y}}(y=1) = \theta_y, \left\{\mu_{ik} \in \mathbb{R}, \sigma_{ik} > 0\right\}_{i=1,k=0}^{m,1} \right\}$$

- **Evaluation**

$$\ln\left(\mathrm{L}(\theta_y)\right) \qquad \text{and} \qquad \left\{ \ln\left(\mathrm{L}(\mu_{ik}, \sigma_{ik})\right) \right\}_{i=1,k=0}^{m,1}$$

- **Optimisation**

$$\theta_{y\mathrm{MLE}} = \frac{n_1}{n} \qquad \text{and} \qquad \left\{ \mu_{ik\mathrm{MLE}} = \sum_{j=1}^{n_k} \frac{x_i^{(j)}}{n_k}, \sigma^2_{ik\mathrm{MLE}} = \frac{1}{n_k} \sum_{j=1}^{n_k} \left(x_i^{(j)} - \mu_{ik}\right)^2 \right\}_{i=1,k=0}^{m,1}$$

Gaussian Naïve Bayes & Logistic Regression

- While we have already motivated **Logistic Regression** we may examine it further and view it as a sort of generalisation of the **GNB** algorithm

- This gives and insight into the tradeoff between **Generative** and **Discriminative** methods more generally

Bayes Rule Revisited

$$
\begin{aligned}
p_{\mathcal{Y}}(y=1|\mathbf{x}) &= \frac{p_{\mathcal{Y}}(y=1)p_{\mathcal{X}}(\mathbf{x}|y=1)}{p_{\mathcal{Y}}(y=1)p_{\mathcal{X}}(\mathbf{x}|y=1)+p_{\mathcal{Y}}(y=0)p_{\mathcal{X}}(\mathbf{x}|y=0)} \\
&= \frac{1}{1+\frac{p_{\mathcal{Y}}(y=0)p_{\mathcal{X}}(\mathbf{x}|y=0)}{p_{\mathcal{Y}}(y=1)p_{\mathcal{X}}(\mathbf{x}|y=1)}} \\
&= \frac{1}{1+\exp\left(\ln\left(\frac{p_{\mathcal{Y}}(y=0)p_{\mathcal{X}}(\mathbf{x}|y=0)}{p_{\mathcal{Y}}(y=1)p_{\mathcal{X}}(\mathbf{x}|y=1)}\right)\right)} \\
&= \frac{1}{1+\exp\left(\ln\left(\frac{p_{\mathcal{Y}}(y=0)}{p_{\mathcal{Y}}(y=1)}\right)+\ln\left(\frac{p_{\mathcal{X}}(\mathbf{x}|y=0)}{p_{\mathcal{X}}(\mathbf{x}|y=1)}\right)\right)}
\end{aligned}
$$

Logistic Regression

- At this point the **Logistic Regression** assumption is to make the following **modelling choice**:

$$\ln\left(\frac{p_{\mathcal{Y}}(y=0)}{p_{\mathcal{Y}}(y=1)}\right) + \ln\left(\frac{p_{\mathcal{X}}(\mathbf{x}|y=0)}{p_{\mathcal{X}}(\mathbf{x}|y=1)}\right) = -\mathbf{w}\cdot\mathbf{x}$$

- And recall that $\mathbf{w}\cdot\mathbf{x} = 0$ defines a **linear discriminant** because:

$$\begin{aligned}
\mathbf{w}\cdot\mathbf{x} \geqslant 0 \quad &\Longrightarrow \quad p_{\mathcal{Y}}(y=1|\mathbf{x}) \geqslant 0.5 \quad &\Longrightarrow \quad f_{\mathbf{w}}(\mathbf{x}) = 1 \\
\mathbf{w}\cdot\mathbf{x} < 0 \quad &\Longrightarrow \quad p_{\mathcal{Y}}(y=0|\mathbf{x}) < 0.5 \quad &\Longrightarrow \quad f_{\mathbf{w}}(\mathbf{x}) = 0
\end{aligned}$$

Naïve Bayes

- On the other hand, Naïve Bayes gives us:

$$\ln\left(\frac{p_{\mathcal{Y}}(y=0)}{p_{\mathcal{Y}}(y=1)}\right) = \ln\left(\frac{1-\theta_y}{\theta_y}\right)$$

- And using the **conditional independence** assumption:

$$\ln\left(\frac{p_{\mathcal{X}}(\mathbf{x}|y=0)}{p_{\mathcal{X}}(\mathbf{x}|y=1)}\right) = \sum_{i=1}^{m} \ln\left(\frac{p_{\mathcal{X}_i}(x_i|y=0)}{p_{\mathcal{X}_i}(x_i|y=1)}\right)$$

UCL

Gaussian Naïve Bayes

- Furthermore, the **Gaussianity** assumption, and an assumption that $\sigma_{ik} = \sigma_i$, implies:

$$
\begin{aligned}
\ln\left(\frac{p_{\mathcal{X}}(\mathbf{x}|y=0)}{p_{\mathcal{X}}(\mathbf{x}|y=1)}\right) &= \sum_{i=1}^{m} \ln\left(\frac{\frac{1}{\sqrt{2\pi\sigma_i^2}}\exp\left(-\frac{(x_i-\mu_{i0})^2}{2\sigma_i^2}\right)}{\frac{1}{\sqrt{2\pi\sigma_i^2}}\exp\left(-\frac{(x_i-\mu_{i1})^2}{2\sigma_i^2}\right)}\right) \\
&= \sum_{i=1}^{m} \ln\left(\exp\left(-\frac{(x_i-\mu_{i0})^2}{2\sigma_i^2} + \frac{(x_i-\mu_{i1})^2}{2\sigma_i^2}\right)\right) \\
&= \sum_{i=1}^{m} \left(\frac{-x_i^2 - \mu_{i0}^2 + 2x_i\mu_{i0} + x_i^2 + \mu_{i1}^2 - 2x_i\mu_{i1}}{2\sigma_i^2}\right) \\
&= \sum_{i=1}^{m} \left(x_i\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2} + \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2}\right)
\end{aligned}
$$

Gaussian Naïve Bayes

- But this means that:

$$\ln\left(\frac{p_{\mathcal{Y}}(y=0)}{p_{\mathcal{Y}}(y=1)}\right)+\ln\left(\frac{p_{\mathcal{X}}(\mathbf{x}|y=0)}{p_{\mathcal{X}}(\mathbf{x}|y=1)}\right)=\ln\left(\frac{1-\theta_y}{\theta_y}\right)+\sum_{i=1}^{m}\left(x_i\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}+\frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}\right)$$
$$=w_0+\sum_{i=1}^{m}w_i x_i$$

- Where:

$w_0=\ln\left(\frac{1-\theta_y}{\theta_y}\right)+\sum_{i=1}^{m}\left(\frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}\right)$

$w_i=\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}$

- In other words we have demonstrated that GNB (with non-class contingent variances) also results in a **linear discriminant**

Gaussian Naïve Bayes & Logistic Regression Compared

- GNB and LR both result in a similar form of **linear discriminant classifier**
- However they do not result in the same classifier:
  - **w** will be different for each method
- GNB makes different model assumptions to LR
- GNB and LR are different **representations**, with different restrictions on how the parameters are set:
  - LR places fewer restrictions on these parameters
  - GNB explicitly defines a dependence between w_0 and w_i

Gaussian Naïve Bayes & Logistic Regression Compared

- Recall that **generative** classifiers require us to learn more parameters than **discriminative** classifiers, *all other things being equal*
- But the generative approach is often intractable, so we are forced to make model assumptions...
- ...This means that a discriminative method such as LR makes fewer model assumptions than a generative one such as NB...
- ...This means that LR is considered more **robust** and less sensitive to modelling choices than NB...
- However, because of these parameter restrictions then (if modelling assumptions are good) NB requires less data than LR for a similar level of convergence in parameter estimates

Lecture Overview

Summary

1. The **Generative** approach to classification is demanding and sometimes intractable. It prompts us to make simplifying assumptions

2. The **Naïve Bayes** algorithm flows from the **conditional independence** assumption.

   - Then depending on the form which we assume for the **class contingent distribution** we are led to different NB algorithms

3. Both **Logistic Regression** and **Gaussian NB** lead to similar forms of **linear classifier**, but may result in very different discriminant boundaries

In the next lecture we will return to more theoretical considerations and discuss the problem of **Model Selection**