Classification – Decision boundary & Naïve Bayes

Sub-lecturer: Mariya Toneva

Instructor: Aarti Singh

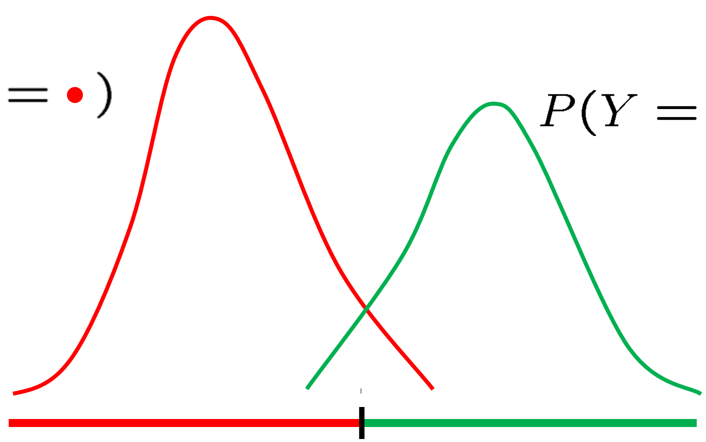
Machine Learning 10-315
Sept 4, 2019

1-dim Gaussian Bayes classifier

$$f^*(x) = \arg\max_{Y=y} \underbrace{P(X = x|Y = y)}_{\text{Class conditional density}} \underbrace{P(Y = y)}_{\text{Class probability}}$$

Gaussian(μ_y, σ^2_y) Bernoulli(θ)

$P(Y = \bullet)P(X = x|Y = \bullet)$



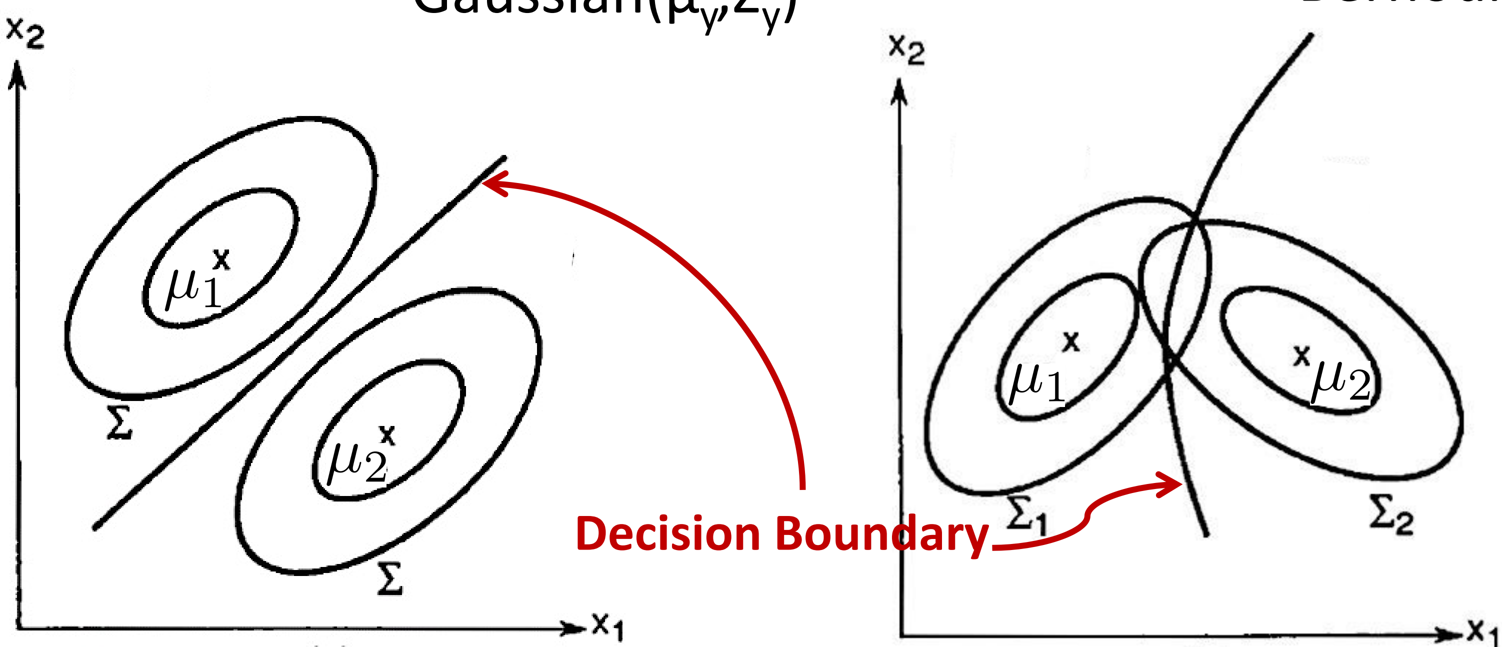
$P(Y = \bullet)P(X = x|Y = \bullet)$

d-dim Gaussian Bayes classifier

$$f^*(x) = \arg\max_{Y=y} \underbrace{P(X=x|Y=y)}_{\text{Class conditional density}} \underbrace{P(Y=y)}_{\text{Class probability}}$$

Gaussian(μ_y,Σ_y)

Bernoulli(θ)



Decision Boundary of Gaussian Bayes

- Decision boundary is set of points x: P(Y=1|X=x) = P(Y=0|X=x)

If class conditional feature distribution P(X=x|Y=y) is 2-dim Gaussian $N(\mu_y,\Sigma_y)$

$$P(X=x|Y=y) = \frac{1}{\sqrt{(2\pi)^d|\Sigma_y|}} \exp\left(-\frac{(x-\mu_y)\Sigma_y^{-1}(x-\mu_y)'}{2}\right)$$

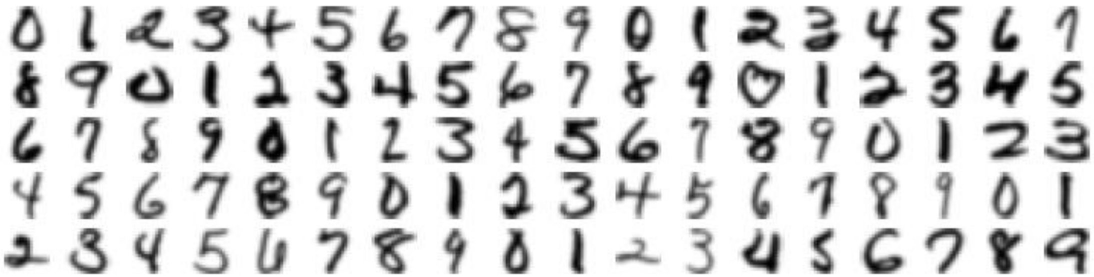
$$\frac{P(Y=1|X=x)}{P(Y=0|X=x)} = \frac{P(X=x|Y=1)P(Y=1)}{P(X=x|Y=0)P(Y=0)}$$

$$= \sqrt{\frac{|\Sigma_0|}{|\Sigma_1|}} \exp\left(- \quad \frac{(x-\mu_1)\Sigma_1^{-1}(x-\mu_1)'}{2} + \frac{(x-\mu_0)\Sigma_0^{-1}(x-\mu_0)'}{2}\right) \frac{\theta}{1-\theta}$$

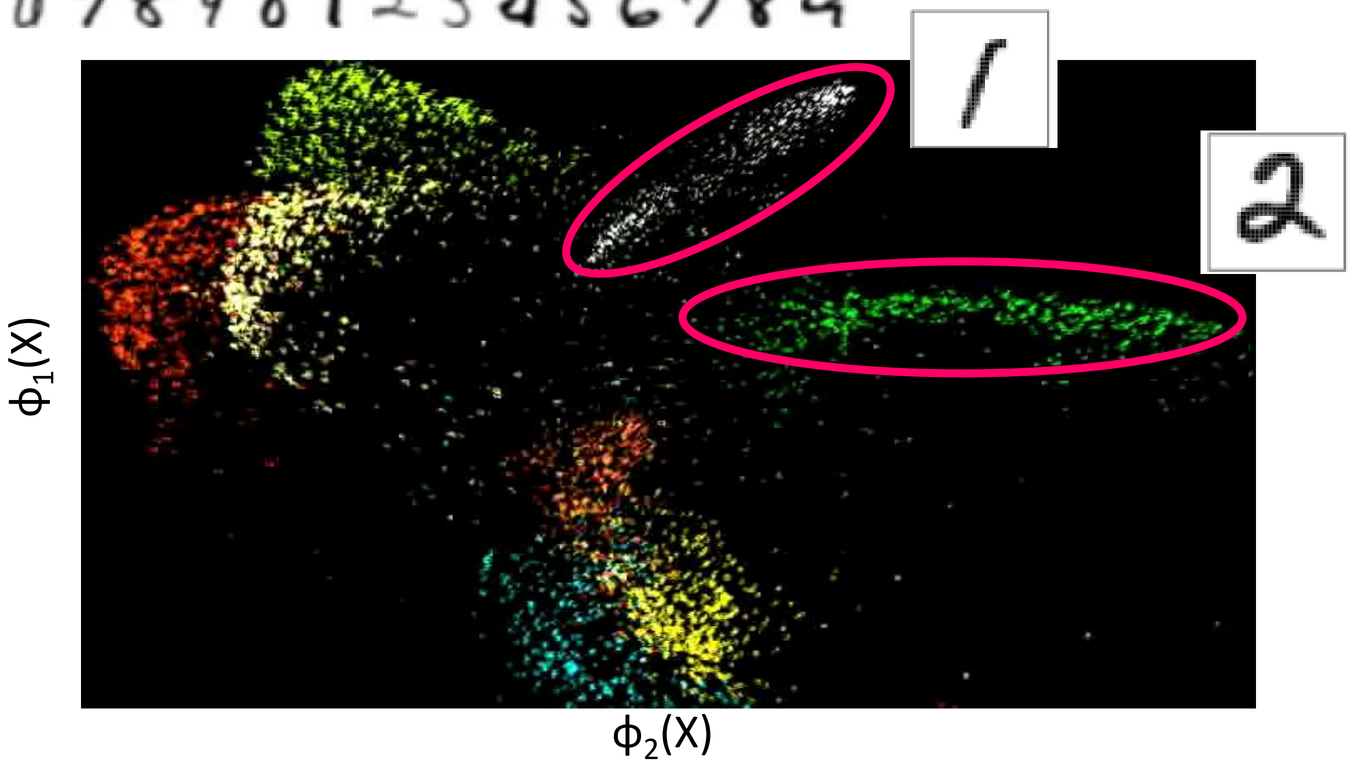
<u>Note</u>: In general, this implies a quadratic equation in x. But if $\Sigma_1 = \Sigma_0$, then quadratic part cancels out and decision boundary is linear.

Multi-class problem
Multi-dimensional input X

Handwritten digit recognition



Multi-class classification

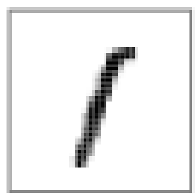
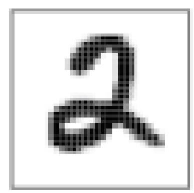


Note: 8 digits shown out of 10 (0, 1, …, 9);
Axes are obtained by nonlinear dimensionality reduction (later in course)

Handwritten digit recognition

Training Data:

Each image represented as a vector of **intensity values at the d pixels (features)**

Input, X   … n greyscale images

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_d \end{bmatrix}$$

Label, Y   1   2   … n labels

Gaussian Bayes model:

$P(Y = y) = p_y$ for all y in 0, 1, 2, …, 9

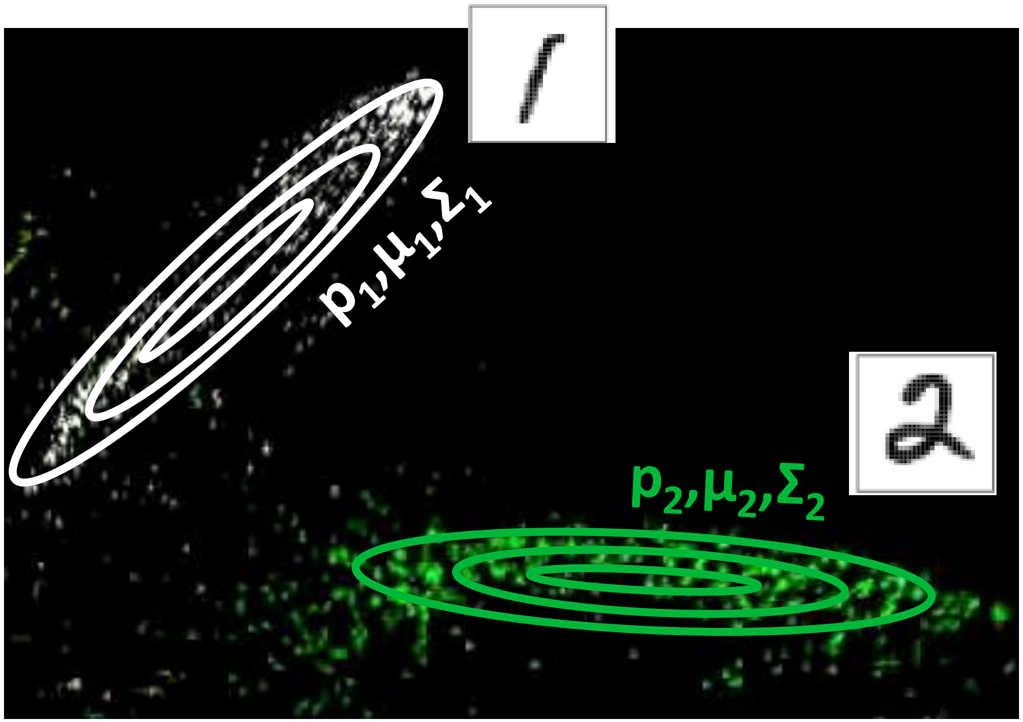
$p_0, p_1, \dots, p_9$ (sum to 1)

$P(X=x|Y = y) \sim N(\mu_y,\Sigma_y)$ for each y

μ_y – d-dim vector

Σ_y - dxd matrix

Gaussian Bayes classifier



How to learn parameters p_y, μ_y, Σ_y from data?

$P(Y = y) = p_y$ for all y in 0, 1, 2, …, 9

$p_0, p_1, \ldots, p_9$ (sum to 1)

$P(X=x|Y = y) \sim N(\mu_y,\Sigma_y)$ for each y

μ_y – d-dim vector

Σ_y - dxd matrix

How many parameters do we need to learn?

Class probability:

$P(Y = y) = p_y$ for all y in 0, 1, 2, …, 9 $\qquad$ $p_0, p_1, \ldots, p_9$ (sum to 1)

K-1 if K labels

Class conditional distribution of features:

$P(X=x|Y = y) \sim N(\mu_y,\Sigma_y)$ for each y $\qquad$ μ_y – d-dim vector

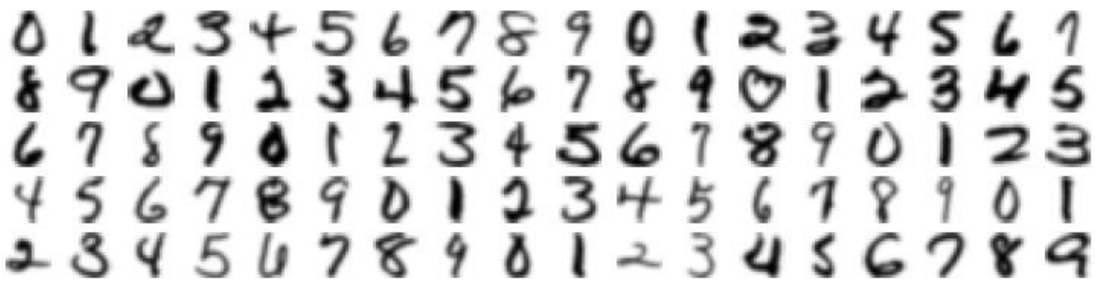
Σ_y - dxd matrix

Kd + Kd(d+1)/2 = O(Kd^2) if d features

Quadratic in dimension d! If d = 256x256 pixels, ~ 21.5 billion parameters!

Hand-written digit recognition

Input, X (images of hand-written digits) → Label, Y



0, 1, 2, …, 9

Feature representation:

Grey-scale images – d-dim vector of d pixel intensities

Continuous features

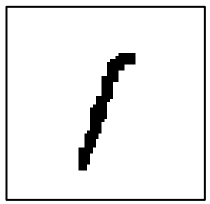
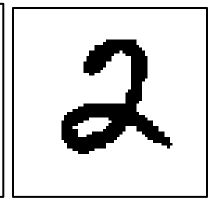
Black-white images – d-dim binary (0/1) vector

Discrete features

What about discrete features?

Each image represented as a vector of **d binary features (black 1 or white 0)**

Training Data:

Input, X   … n **black-white** images

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_d \end{bmatrix}$$

Label, Y 1 2 … n labels

Discrete Bayes model:

$P(Y = y) = p_y$ for all y in 0, 1, 2, …, 9 $p_0, p_1, \ldots, p_9$ (sum to 1)

$P(X=x|Y = y)$ ~ For each label y, maintain probability table with 2^d-1 entries

How many parameters do we need to learn?

Class probability:

$P(Y = y) = p_y$ for all y in 0, 1, 2, …, 9 $p_0, p_1, \ldots, p_9$ (sum to 1)

K-1 if K labels

Class conditional distribution of features:

$P(X=x|Y = y)$ ~ For each label y, maintain probability table with 2^d-1 entries

$K(2^d – 1)$ if d binary features

Exponential in dimension d!

What's wrong with too many parameters?

- How many training data needed to learn one parameter (bias of a coin)?

- Need lots of training data to learn the parameters!
  - Training data > number of parameters

Naïve Bayes Classifier

- Bayes Classifier with additional "naïve" assumption:
  - Features are independent given class:

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

$$\begin{aligned} P(X_1, X_2|Y) &= P(X_1|X_2, Y)P(X_2|Y) \\ &= P(X_1|Y)P(X_2|Y) \end{aligned}$$

  - More generally:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_d \end{bmatrix}$$

$$P(X_1 ... X_d|Y) = \prod_{i=1}^{d} P(X_i|Y)$$

- If conditional independence assumption holds, NB is optimal classifier! But worse otherwise.

Conditional Independence

- X is **conditionally independent** of Y given Z:

  probability distribution governing X is independent of the value of Y, given the value of Z

$$(\forall x, y, z) P(X = x | Y = y, Z = z) = P(X = x | Z = z)$$

- Equivalent to:

$$P(X, Y \mid Z) = P(X \mid Z) P(Y \mid Z)$$

- e.g., $P(Thunder | Rain, Lightning) = P(Thunder | Lightning)$

  Note: does NOT mean Thunder is independent of Rain

Conditional vs. Marginal Independence

London taxi drivers: A survey has pointed out a positive and significant correlation between the number of accidents and wearing coats. They concluded that coats could hinder movements of drivers and be the cause of accidents. A new law was prepared to prohibit drivers from wearing coats when driving.

Finally another study pointed out that people wear coats when it rains…

Wearing coats is independent of accidents conditioning on the fact that it rained

Naïve Bayes Classifier

- Bayes Classifier with additional "naïve" assumption:
  - Features are independent given class:

$$P(X_1 \ldots X_d | Y) = \prod_{i=1}^{d} P(X_i | Y)$$

$$\begin{aligned} f_{NB}(\mathbf{x}) &= \arg\max_y \; P(x_1, \ldots, x_d \mid y) P(y) \\ &= \arg\max_y \prod_{i=1}^{d} P(x_i | y) P(y) \end{aligned}$$

- How many parameters now?

Handwritten digit recognition (continuous features)

Training Data:

$$\mathrm{X} = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_d \end{bmatrix}$$


… n greyscale images with d pixels

Y 1 2 … n labels

How many parameters?

Class probability $P(Y = y) = p_y$ for all y **K-1 if K labels**

May not hold

Class conditional distribution of features (using Naïve Bayes assumption)

$P(X_i = x_i \mid Y = y) \sim N(\mu^{(y)}_i, \sigma^2_i{}^{(y)})$ for each y and each pixel i **2Kd**

Linear instead of Quadratic in d!

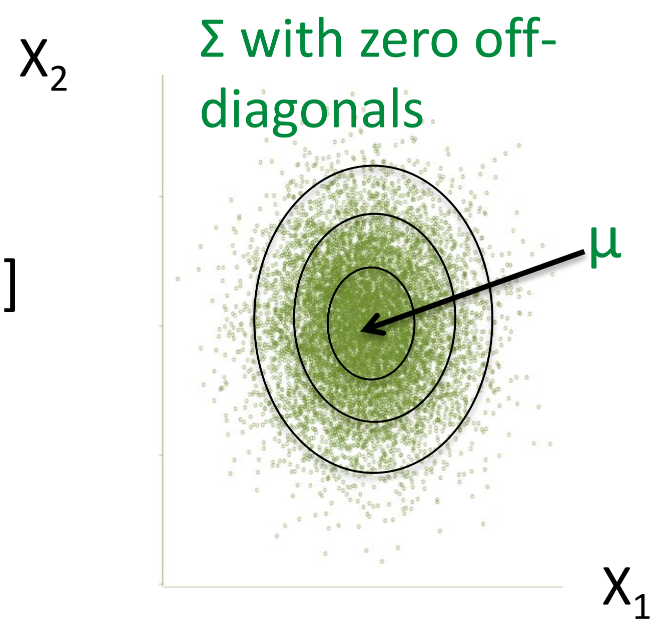
Independent Gaussians

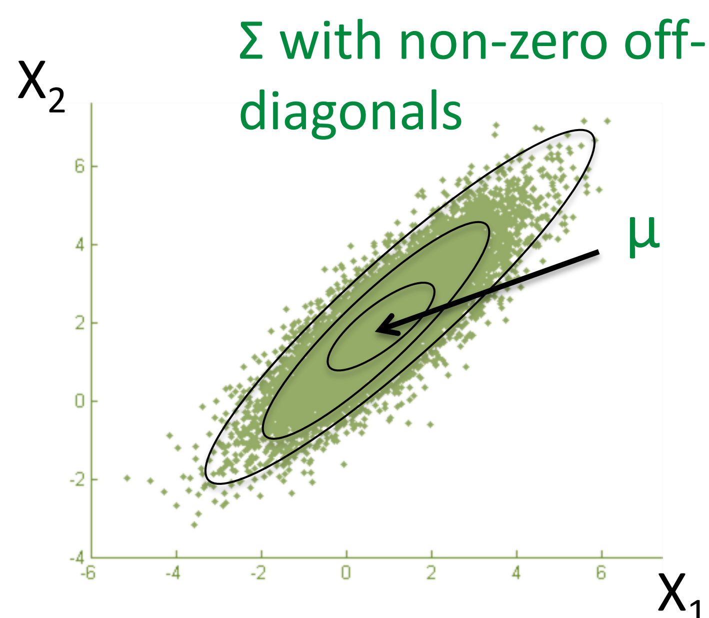
Equivalent to assuming

$$\Sigma_y = \begin{bmatrix} \sigma_1^{2(y)} & 0 & 0 & 0 \\ 0 & \sigma_2^{2(y)} & 0 & 0 \\ & & \cdots & \\ 0 & 0 & 0 & \sigma_d^{2(y)} \end{bmatrix}$$

d=2

$X = [X_1; X_2]$

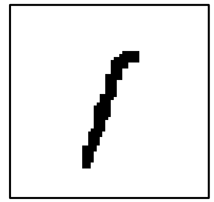


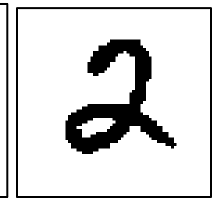


Handwritten digit recognition (discrete features)

Training Data:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_d \end{bmatrix}$$





… n black-white (1/0) images with d pixels

Y 1 2 … n labels

How many parameters?

Class probability $P(Y = y) = p_y$ for all y **K-1 if K labels**

May not hold

Class conditional distribution of features (using Naïve Bayes assumption)

$P(X_i = x_i | Y = y)$ – one probability value for each y, pixel i **Kd**

Linear instead of Exponential in d!

Naïve Bayes Classifier

- Bayes Classifier with additional "naïve" assumption:
  - Features are independent given class:

$$P(X_1 \ldots X_d | Y) = \prod_{i=1}^{d} P(X_i | Y)$$

$$\begin{aligned} f_{NB}(\mathbf{x}) &= \arg\max_y \; P(x_1, \ldots, x_d \mid y) P(y) \\ &= \arg\max_y \prod_{i=1}^{d} P(x_i | y) P(y) \end{aligned}$$

- Has fewer parameters, and hence requires fewer training data, even though assumption may be violated in practice

How to learn parameters from data? MLE, MAP

(Discrete case)

Learning parameters in distributions

$$P(Y = \color{red}{\bullet}) = \theta \qquad\qquad P(Y = \color{green}{\bullet}) = 1 - \theta$$

Learning θ is equivalent to learning probability of head in coin flip.

How do you learn that?

Data =

Answer: 3/5

Why??

Bernoulli distribution

Data, D = 

- P(Heads) = θ, P(Tails) = 1-θ
- Flips are **i.i.d.**:
  - **Independent** events
  - **Identically distributed** according to Bernoulli distribution

Choose θ that maximizes the probability of observed data

Maximum Likelihood Estimation (MLE)

Choose θ that maximizes the probability of observed data (aka likelihood)

$$\widehat{\theta}_{MLE} = \arg\max_{\theta} P(D \mid \theta)$$

MLE of probability of head:

$$\widehat{\theta}_{MLE} = \frac{\alpha_H}{\alpha_H + \alpha_T}$$ **= 3/5**

"Frequency of heads"

Multinomial distribution

Data, D = rolls of a dice

- $P(1) = p_1$, $P(2) = p_2$, …, $P(6) = p_6$ $\quad\quad p_1 + \ldots + p_6 = 1$
- Rolls are **i.i.d.**:
  - **Independent** events
  - **Identically distributed** according to Multinomial(θ) distribution where $\theta = \{p_1, p_2, \ldots, p_6\}$

Choose θ that maximizes the probability of observed data

Maximum Likelihood Estimation (MLE)

Choose θ that maximizes the probability of observed data

$$\widehat{\theta}_{MLE} = \arg\max_{\theta} \; P(D \mid \theta)$$

MLE of probability of rolls:

$$\widehat{\theta}_{MLE} = \hat{p}_{1,MLE}, \ldots, \hat{p}_{6,MLE}$$

$$\hat{p}_{y,MLE} = \frac{\alpha_y}{\sum_y \alpha_y}$$

Rolls that turn up y

Total number of rolls

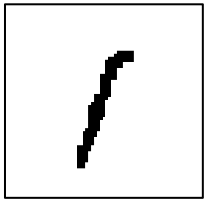
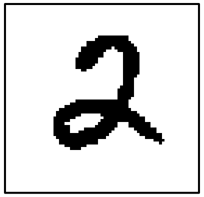
"Frequency of roll y"

Back to Naïve Bayes

Naïve Bayes with discrete features

Training Data:

Each image represented as a vector of **d binary features (black 1 or white 0)**

Input, X   … n **black-white** images

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_d \end{bmatrix}$$

Label, Y 1 2 … n labels

Discrete Naïve Bayes model:

$P(Y = y) = p_y$ for all y in 0, 1, 2, …, 9 $p_0, p_1, \dots, p_9$ (sum to 1)

$P(X_i = x_i \mid Y = y)$ - one probability value for each y, pixel i

Naïve Bayes Algo – Discrete features

- Training Data $\{(X^{(j)}, Y^{(j)})\}_{j=1}^{n}$ $\quad X^{(j)} = (X_1^{(j)}, \ldots, X_d^{(j)})$
- Maximum Likelihood Estimates
  - For Class probability

$$\widehat{P}(y) = \frac{\{\#j : Y^{(j)} = y\}}{n}$$

  - For class conditional distribution

$$\frac{\widehat{P}(x_i, y)}{\widehat{P}(y)} = \frac{\{\#j : X_i^{(j)} = x_i, Y^{(j)} = y\}/n}{\{\#j : Y^{(j)} = y\}/n}$$

- NB Prediction for test data $\quad X = (x_1, \ldots, x_d)$

$$Y = \arg\max_y \widehat{P}(y) \prod_{i=1}^{d} \frac{\widehat{P}(x_i, y)}{\widehat{P}(y)}$$

Issues with Naïve Bayes

- **Issue 1:** Usually, features are not conditionally independent:

$$P(X_1 ... X_d|Y) \neq \prod_i P(X_i|Y)$$

  Nonetheless, NB is the single most used classifier particularly when data is limited, works well

- **Issue 2:** Typically use MAP estimates instead of MLE since insufficient data may cause MLE to be zero.

Insufficient data for MLE

- What if you never see a training instance where X_1=a when Y=b?
  - e.g., b={SpamEmail}, a ={‘Earn’}
  - $P(X_1 = a \mid Y = b) = 0$
- Thus, no matter what the values $X_2,\ldots,X_d$ take:

$$\boxed{\widehat{P}(X_1 = a, X_2 \ldots X_n | Y) = \widehat{P}(X_1 = a|Y) \prod_{i=2}^{d} \widehat{P}(X_i|Y)} = 0$$

- What now???