School of Computer Science
Carnegie Mellon

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 $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$

$$\boldsymbol{\theta}^{\text{MLE}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)} | \boldsymbol{\theta})$$

Maximum Likelihood Estimate (MLE)

Background: MLE

Example: MLE of Exponential Distribution

- pdf of Exponential(λ): $f(x) = \lambda e^{-\lambda x}$
- Suppose $X_i \sim$ Exponential(λ) 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 λ.
- Compute second derivative and check that it is concave down at λ^{MLE}.

Background: MLE

Example: MLE of Exponential Distribution

- First write down log-likelihood of sample.

$$
\begin{aligned}
\ell(\lambda) &= \sum_{i=1}^{N} \log f(x^{(i)}) && (1)\\
&= \sum_{i=1}^{N} \log(\lambda \exp(-\lambda x^{(i)})) && (2)\\
&= \sum_{i=1}^{N} \log(\lambda) + -\lambda x^{(i)} && (3)\\
&= N \log(\lambda) - \lambda \sum_{i=1}^{N} x^{(i)} && (4)
\end{aligned}
$$

Background: MLE

Example: MLE of Exponential Distribution

- Compute first derivative, set to zero, solve for λ.

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

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

$$\boxed{\Rightarrow \lambda^{\text{MLE}} = \frac{N}{\sum_{i=1}^{N} x^{(i)}}} \tag{3}$$

Background: MLE

Example: MLE of Exponential Distribution

- pdf of Exponential(λ): $f(x) = \lambda e^{-\lambda x}$
- Suppose $X_i \sim$ Exponential(λ) 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 λ.
- Compute second derivative and check that it is concave down at λ^{MLE}.

MLE vs. MAP

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

Maximum Likelihood Estimate (MLE)

$$\boldsymbol{\theta}^{\text{MLE}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)} | \boldsymbol{\theta})$$

Maximum *a posteriori* (MAP) estimate

$$\boldsymbol{\theta}^{\text{MAP}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)} | \boldsymbol{\theta}) \underbrace{p(\boldsymbol{\theta})}_{\text{Prior}}$$

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)

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

Option 2: Binary vector (word indicators)

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

Option 3: Integer vector (word counts)

$\mathbf{x} = [x_1, x_2, \ldots, x_K]$ where $x_k \in \mathbb{Z}^+$ 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

All latest updates

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

Sep 5th 2016 | MADRID | Europe

Timekeeper · Like 80 · Tweet

BACK in June, after Spain's second indecisive election in six months, the general expectation was that Mariano Rajoy, the prime minister, would swiftly form a new government. Although his conservative People's Party (PP) did not win back the absolute majority it had lost in December, it remained easily the largest party, with 137 of the 350

The Onion

★ ELECTION 2016 ★ MORE ELECTION COVERAGE ▸

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

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 ($\boldsymbol{X}$) from the Spam distribution
3. If tails, roll the **blue** many sided die to sample a document vector ($\boldsymbol{X}$) 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:

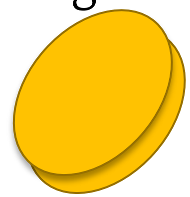
1. Flip a weighted coin (Y)
2. If heads, sample a document ID (X) from the Spam distribution
3. If tails, sample a document ID (X) from the Not-Spam distribution

$$P(X, Y) = P(X|Y)P(Y)$$

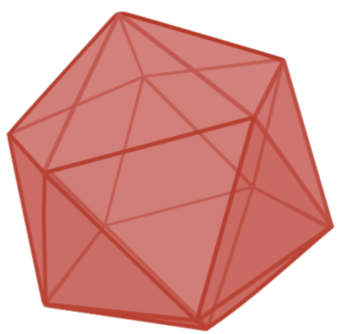
This model is computationally naïve!

Model 0: Not-so-naïve Model?

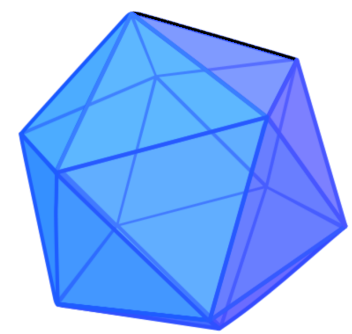
Flip weighted coin



If HEADS, roll red die



If TAILS, roll blue die



y	x_1	x_2	x_3	…	x_K
0	1	0	1	…	1
1	0	1	0	…	1
1	1	1	1	…	1
0	0	0	1	…	1
0	1	0	1	…	0
1	1	0	1	…	0

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

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:

$$
\begin{aligned}
P(X_1, \ldots, X_K, Y) &= P(X_1, \ldots, X_K | Y) P(Y) \\
&= \left(\prod_{k=1}^{K} P(X_k | Y) \right) P(Y)
\end{aligned}
$$

Estimating a joint from conditional probabilities

C	P(C)
0	0.33
1	0.67

$$P(A,B \mid C) = P(A \mid C) * P(B \mid C)$$

$$\forall a,bc : P(A=a \wedge B=b \mid C=c) = P(A=a \mid C=c) * P(B=b \mid C=c)$$

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

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

A	B	C	P(A,B,C)
0	0	0	…
0	0	1	…
0	1	0	…
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Estimating a joint from conditional probabilities

C	P(C)
0	0.33
1	0.67

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

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

A	B	D	C	P(A,B,D,C)
0	0	0	0	…
0	0	1	0	…
0	1	0	0	…
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	0	
0	0	0	1	
0	0	1	0	
…	…	..	…	…

Assuming conditional independence, the conditional probabilities encode the **same information** as the joint table.

They are very convenient for estimating

$P(X_1,\ldots,X_n|Y)=P(X_1|Y)*\ldots*P(X_n|Y)$

They are almost as good for computing

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

$$\forall \mathbf{x},y : P(Y=y|X_1,\ldots,X_n=\mathbf{x})=\frac{P(X_1,\ldots,X_n=\mathbf{x}|Y)P(Y=y)}{P(X_1,\ldots,X_n=\mathbf{x})}$$

Generic Naïve Bayes Model

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

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

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

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

$$\hat{y} = \operatorname*{argmax}_y p(y|\mathbf{x})$$

Generic Naïve Bayes Model

Classification:

$$\hat{y} = \operatorname*{argmax}_y p(y|\mathbf{x}) \quad \text{(posterior)}$$

$$= \operatorname*{argmax}_y \frac{p(\mathbf{x}|y)p(y)}{p(x)} \quad \text{(by Bayes' rule)}$$

$$= \operatorname*{argmax}_y p(\mathbf{x}|y)p(y)$$

Model 1: Bernoulli Naïve Bayes

Support: Binary vectors of length K

$$\mathbf{x} \in \{0,1\}^K$$

Generative Story:

$$Y \sim \text{Bernoulli}(\phi)$$

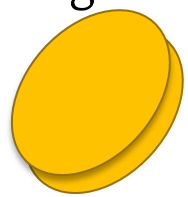
$$X_k \sim \text{Bernoulli}(\theta_{k,Y}) \ \forall k \in \{1,\ldots,K\}$$

Model:

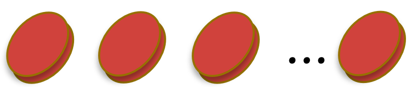
$$\begin{aligned} p_{\phi,\boldsymbol{\theta}}(\boldsymbol{x}, y) &= p_{\phi,\boldsymbol{\theta}}(x_1, \ldots, x_K, y) \\ &= p_{\phi}(y) \prod_{k=1}^{K} p_{\boldsymbol{\theta}_k}(x_k|y) \\ &= (\phi)^y (1-\phi)^{(1-y)} \prod_{k=1}^{K} (\theta_{k,y})^{x_k} (1-\theta_{k,y})^{(1-x_k)} \end{aligned}$$

Model 0: Not-so-naïve Model?

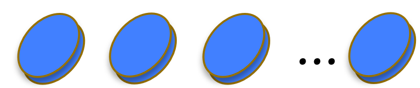
Flip weighted coin



If HEADS, flip each red coin



If TAILS, flip each blue coin



y	x_1	x_2	x_3	…	x_K
0	1	0	1	…	1
1	0	1	0	…	1
1	1	1	1	…	1
0	0	0	1	…	1
0	1	0	1	…	0
1	1	0	1	…	0

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).

Each red coin corresponds to an x_k

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: $p_{\phi,\boldsymbol{\theta}}(\boldsymbol{x}, y) = (\phi)^y (1 - \phi)^{(1-y)} \prod_{k=1}$

Same as Generic Naïve Bayes

Classification: Find the class that maximizes the posterior

$$\hat{y} = \underset{y}{\operatorname{argmax}} \, p(y|\mathbf{x})$$

Generic Naïve Bayes Model

Recall...

Classification:

$$\hat{y} = \operatorname*{argmax}_y p(y|\mathbf{x}) \quad \text{(posterior)}$$

$$= \operatorname*{argmax}_y \frac{p(\mathbf{x}|y)p(y)}{p(x)} \quad \text{(by Bayes' rule)}$$

$$= \operatorname*{argmax}_y p(\mathbf{x}|y)p(y)$$

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.

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

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

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

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

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.

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

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

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

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

Data:

y	x_1	x_2	x_3	…	x_K
0	1	0	1	…	1
1	0	1	0	…	1
1	1	1	1	…	1
0	0	0	1	…	1
0	1	0	1	…	0
1	1	0	1	…	0

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 $p(x_k \mid y)$?

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

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

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

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 we derive the MLEs.

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

$$\mathcal{D}' = \mathcal{D} \cup \{(\mathbf{0}, 0), (\mathbf{0}, 1), (\mathbf{1}, 0), (\mathbf{1}, 1)\} \tag{2}$$

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 $\mathcal{D}$?

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

$$\theta_{k,0} = \frac{\boxed{1 +} \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \wedge x_k^{(i)} = 1)}{\boxed{2 +} \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0)}$$

$$\theta_{k,1} = \frac{\boxed{1 +} \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1 \wedge x_k^{(i)} = 1)}{\boxed{2 +} \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1)}$$

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

2. Add-λ 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-λ smoothing, we add pseudo-observations as before to obtain a smoothed estimate:

$$\phi_k = \frac{\lambda + \sum_{i=1}^{N} \mathbb{I}(x^{(i)} = k)}{k\lambda + N}$$

Recall...

MLE vs. MAP

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

Maximum Likelihood Estimate (MLE)

$$\boldsymbol{\theta}^{\text{MLE}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)} | \boldsymbol{\theta})$$

Maximum *a posteriori* (MAP) estimate

$$\boldsymbol{\theta}^{\text{MAP}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)} | \boldsymbol{\theta}) \underbrace{p(\boldsymbol{\theta})}_{\text{Prior}}$$

3. MAP Estimation (Beta Prior)

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

for $k \in \{1, \ldots, K\}$**:**
  for $y \in \{0, 1\}$**:**
    $\theta_{k,y} \sim \text{Beta}(\alpha, \beta)$
for $i \in \{1, \ldots, N\}$**:**
  $y^{(i)} \sim \text{Bernoulli}(\phi)$
  for $k \in \{1, \ldots, K\}$**:**
    $x_k^{(i)} \sim \text{Bernoulli}(\theta_{k,y^{(i)}})$

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

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

$$\theta_{k,0} = \frac{(\alpha - 1) + \sum_{i=1}^N \mathbb{I}(y^{(i)} = 0 \wedge x_k^{(i)} = 1)}{(\alpha - 1) + (\beta - 1) + \sum_{i=1}^N \mathbb{I}(y^{(i)} = 0)}$$

$$\theta_{k,1} = \frac{(\alpha - 1) + \sum_{i=1}^N \mathbb{I}(y^{(i)} = 1 \wedge x_k^{(i)} = 1)}{(\alpha - 1) + (\beta - 1) + \sum_{i=1}^N \mathbb{I}(y^{(i)} = 1)}$$

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

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)

Model 2: Multinomial Naïve Bayes

Support: Option 1: Integer vector (word IDs)

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

Generative Story:

for $i \in \{1, \dots, N\}$**:**

$y^{(i)} \sim$ Bernoulli(ϕ)

for $j \in \{1, \dots, M_i\}$**:**

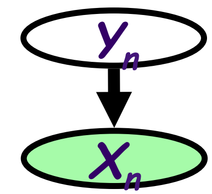
$x_j^{(i)} \sim$ Multinomial($\boldsymbol{\theta}_{y^{(i)}}, 1$)

Model:

$$\begin{aligned} p_{\phi,\boldsymbol{\theta}}(\boldsymbol{x}, y) &= p_\phi(y) \prod_{k=1}^{K} p_{\boldsymbol{\theta}_k}(x_k | y) \\ &= (\phi)^y (1-\phi)^{(1-y)} \prod_{j=1}^{M_i} \theta_{y, x_j} \end{aligned}$$

Gaussian Discriminative Analysis

- learning f: X → Y, where
  - X is a vector of real-valued features, $\mathbf{X}_n = < X_n^1, \ldots X_n^m >$
  - 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:



$$p(\mathbf{x}_n, y_n^k = 1 \mid \mu, \sigma) = p(y_n^k = 1) \times p(\mathbf{x}_n \mid y_n^k = 1, \mu, \Sigma)$$

$$= \pi_k \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\tfrac{1}{2}(\mathbf{x}_n - \vec{\mu}_k)^T \Sigma^{-1} (\mathbf{x}_n - \vec{\mu}_k)\right\}$$

  - Given a datum $\mathbf{x}_n$, we predict its label using the conditional probability of the label given the datum:

$$p(y_n^k = 1 \mid \mathbf{x}_n, \mu, \Sigma) = \frac{\pi_k \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\tfrac{1}{2}(\mathbf{x}_n - \vec{\mu}_k)^T \Sigma^{-1} (\mathbf{x}_n - \vec{\mu}_k)\right\}}{\sum_{k'} \pi_{k'} \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\tfrac{1}{2}(\mathbf{x}_n - \vec{\mu}_k)^T \Sigma^{-1} (\mathbf{x}_n - \vec{\mu}_{k'})\right\}}$$

Model 3: Gaussian Naïve Bayes

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

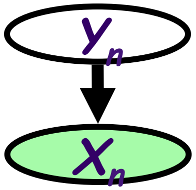
Model: Product of **prior** and the event model

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

Gaussian Naive Bayes assumes that $p(x_k|y)$ is given by a Normal distribution.

Gaussian Naïve Bayes Classifier

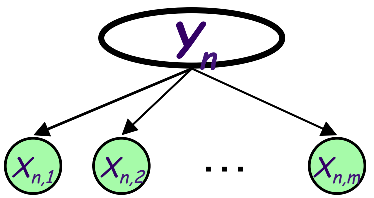
- When X is multivariate-Gaussian vector:
  - The joint probability of a datum and it label is:



$$p(\mathbf{x}_n, y_n^k = 1 \mid \vec{\mu}, \Sigma) = p(y_n^k = 1) \times p(\mathbf{x}_n \mid y_n^k = 1, \vec{\mu}, \Sigma)$$

$$= \pi_k \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{ -\tfrac{1}{2} (\mathbf{x}_n - \vec{\mu}_k)^T \Sigma^{-1} (\mathbf{x}_n - \vec{\mu}_k) \right\}$$

  - The naïve Bayes simplification



$$p(\mathbf{x}_n, y_n^k = 1 \mid \mu, \sigma) = p(y_n^k = 1) \times \prod_j p(x_n^j \mid y_n^k = 1, \mu_k^j, \sigma_k^j)$$

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

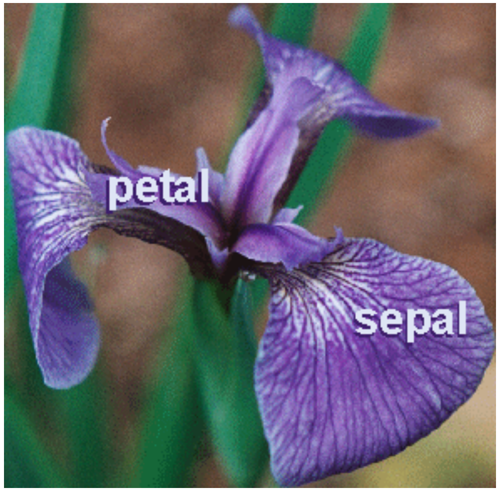
  - More generally:

$$p(\mathbf{x}_n, y_n \mid \eta, \pi) = p(y_n \mid \pi) \times \prod_{j=1}^{m} p(x_n^j \mid y_n, \eta)$$

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

VISUALIZING NAÏVE BAYES

Slides in this section from William Cohen (10-601B, Spring 2016)


petal
sepal

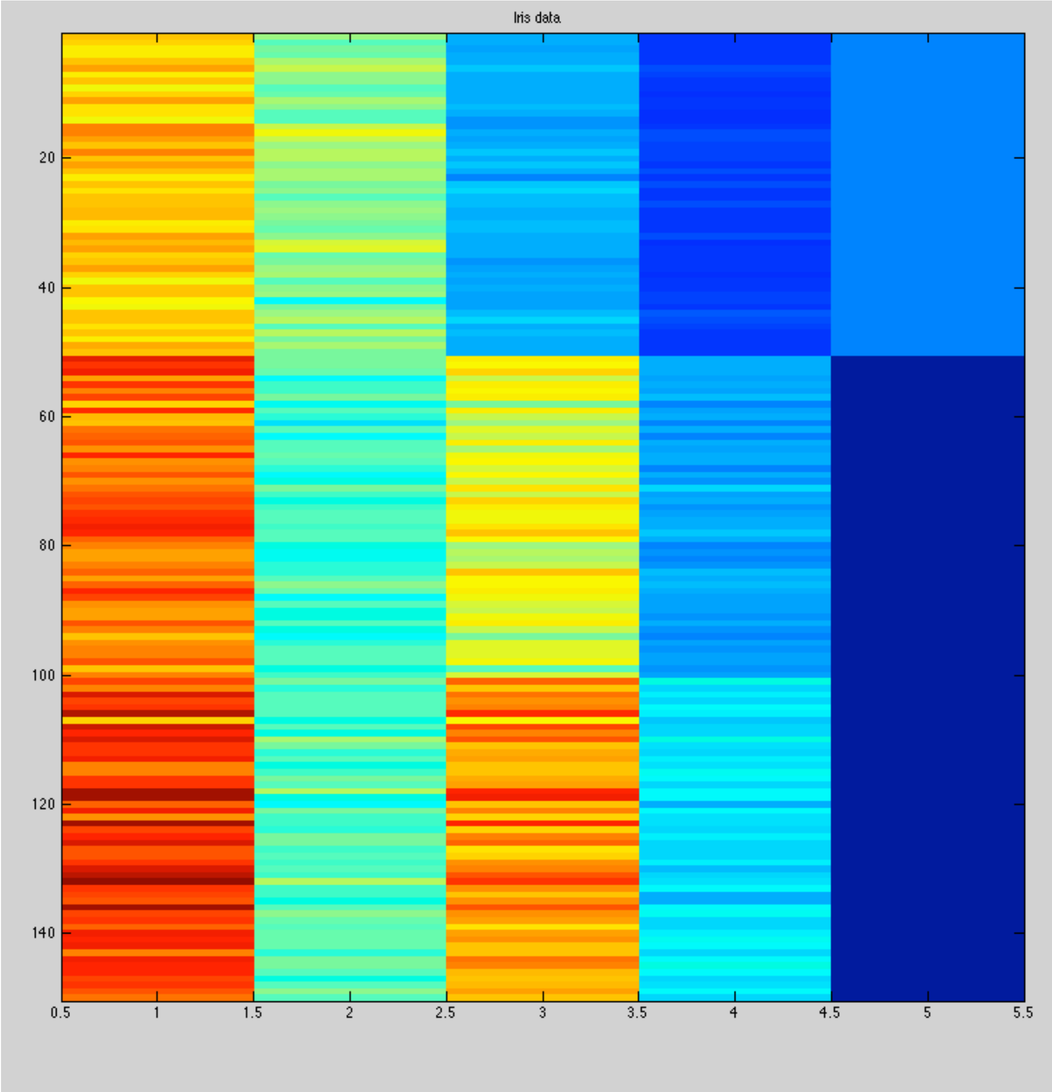
Fisher Iris Dataset

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

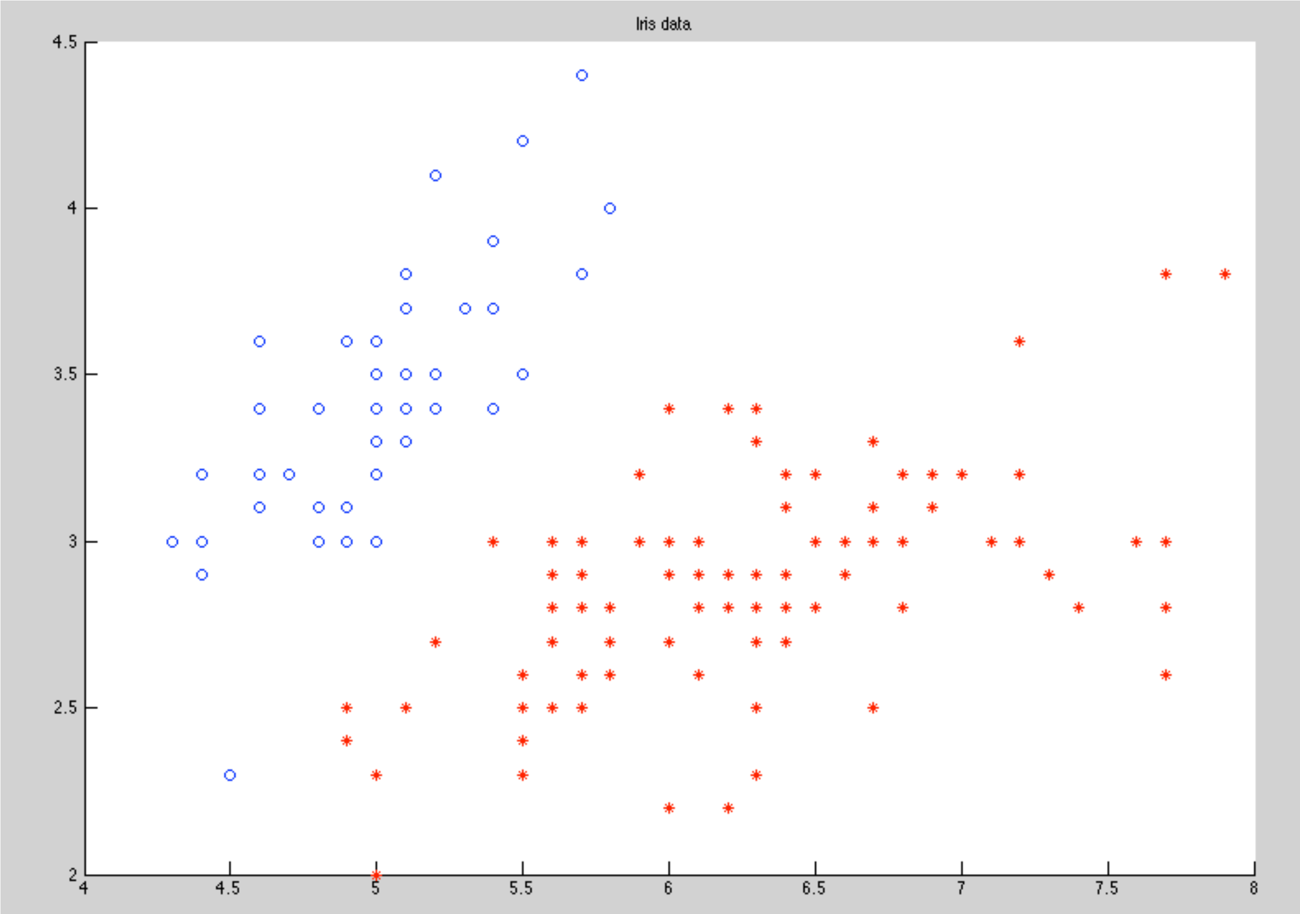
Species	Sepal Length	Sepal Width	Petal Length	Petal Width
0	4.3	3.0	1.1	0.1
0	4.9	3.6	1.4	0.1
0	5.3	3.7	1.5	0.2
1	4.9	2.4	3.3	1.0
1	5.7	2.8	4.1	1.3
1	6.3	3.3	4.7	1.6
1	6.7	3.0	5.0	1.7

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');
```

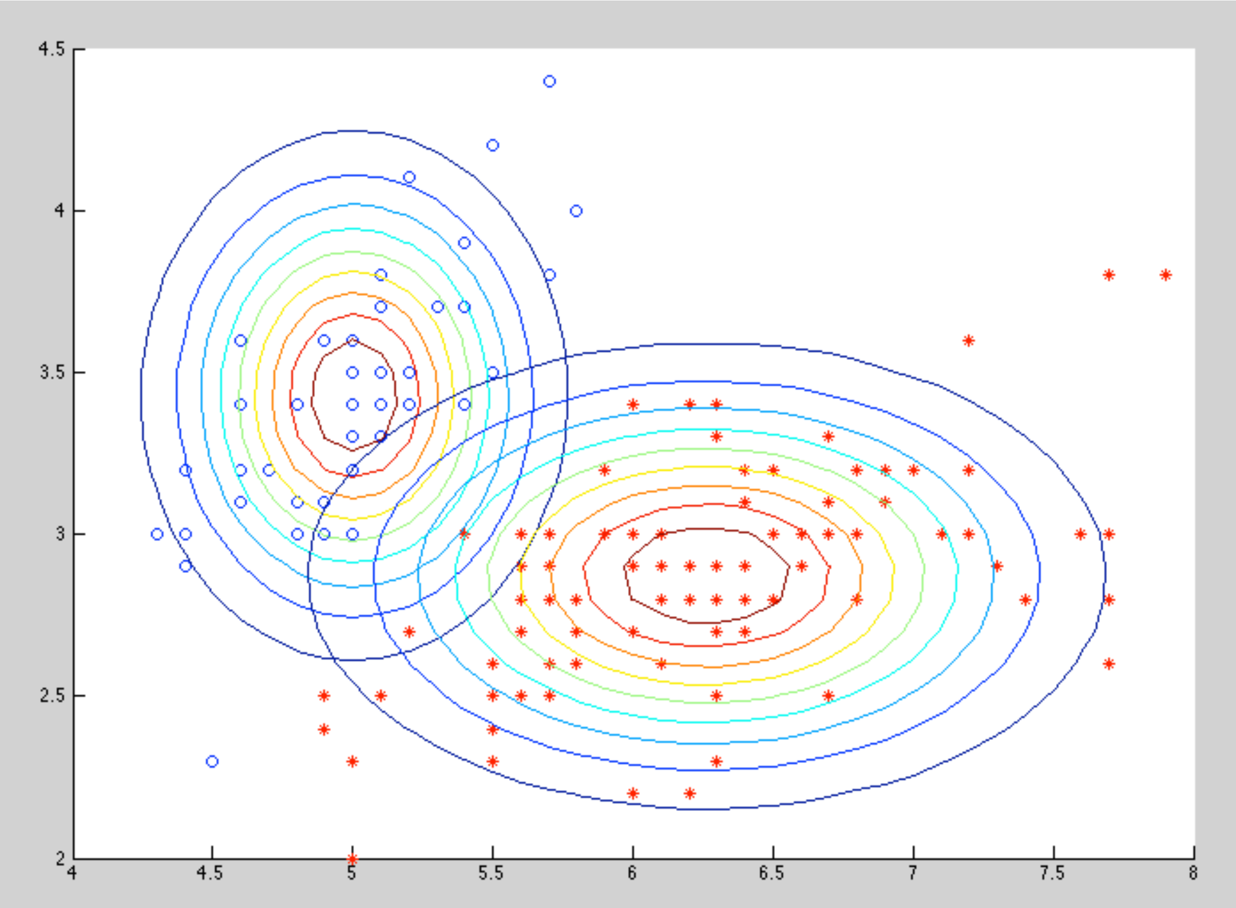


```
%% Visualize by scatter plotting the the first two dimensions
figure;
scatter(X(Y<0,1),X(Y<0,2),'r*');
hold on;
scatter(X(Y>0,1),X(Y>0,2),'bo');
title('Iris data');
```

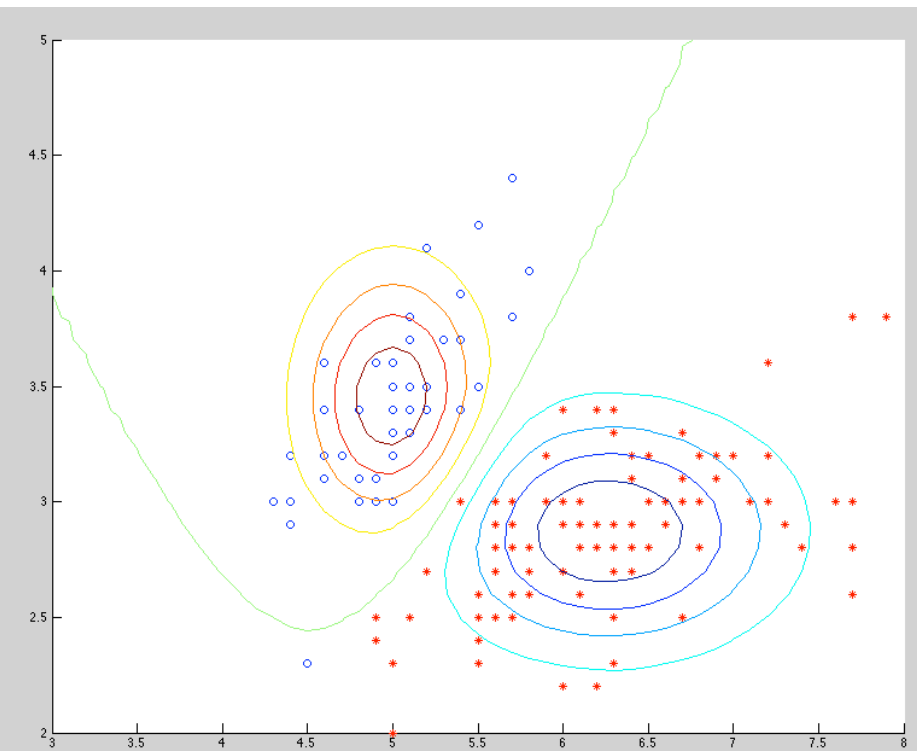


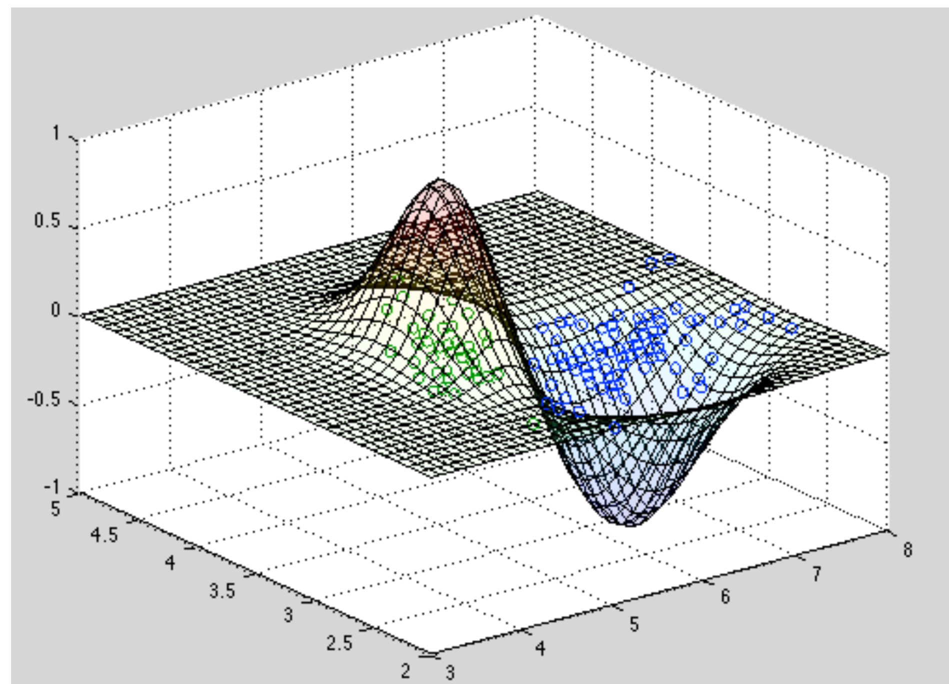
```
%% Compute the mean and SD of each class
PosMean = mean(X(Y>0,:));
PosSD = std(X(Y>0,:));
NegMean = mean(X(Y<0,:));
NegSD = std(X(Y<0,:));

%% 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;
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)
```

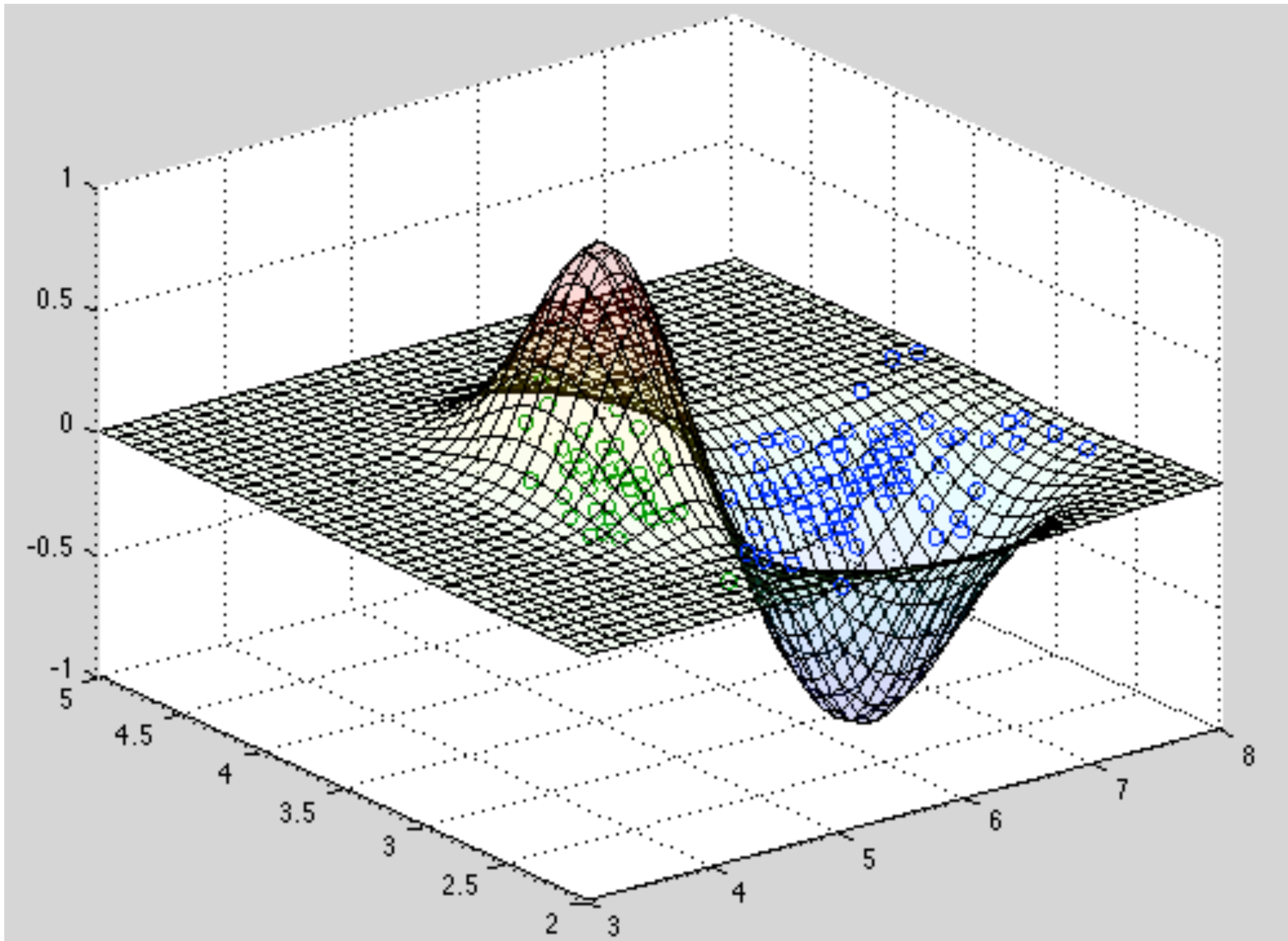


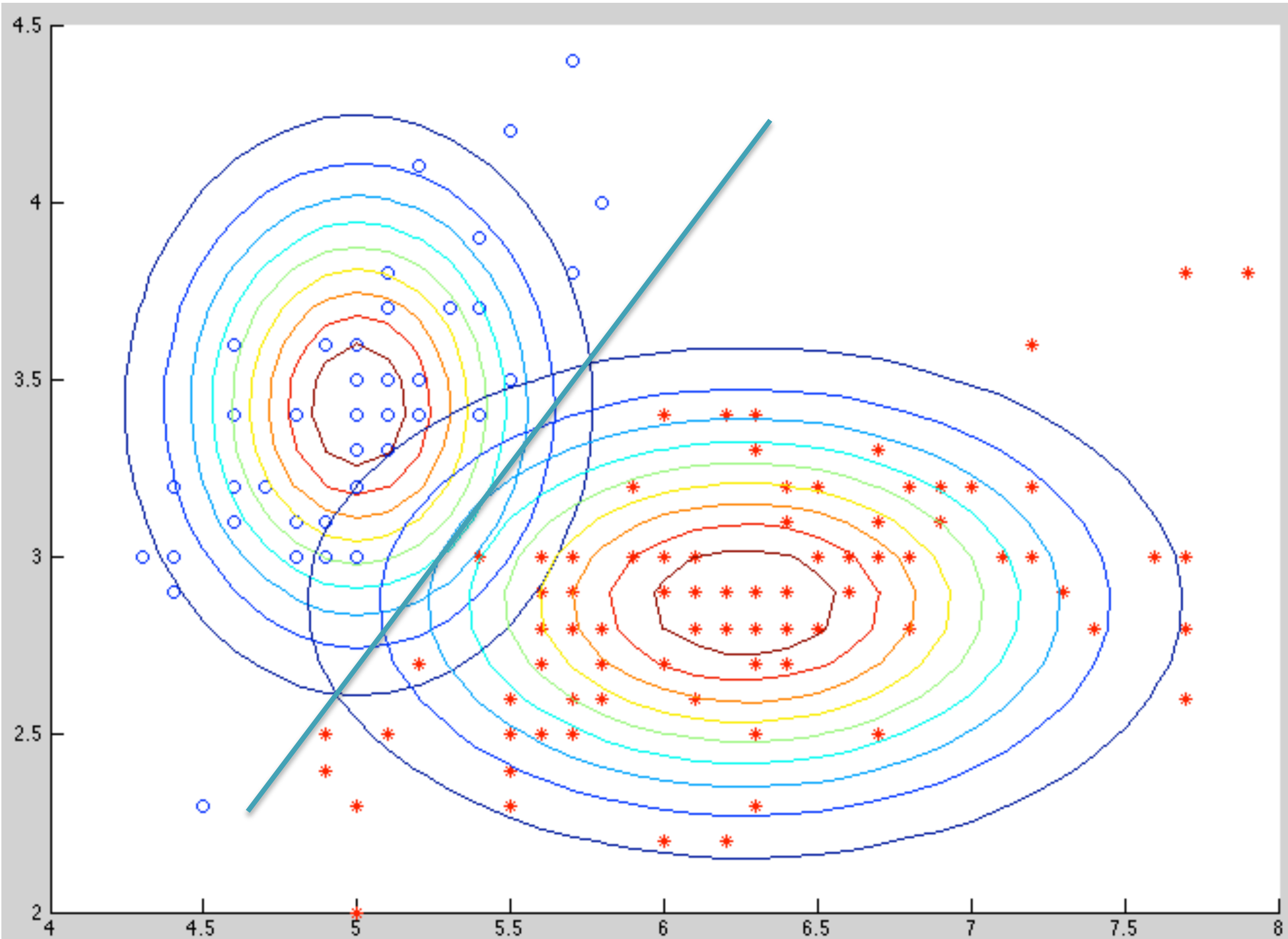


NAÏVE BAYES IS LINEAR

Slides in this section from William Cohen (10-601B, Spring 2016)

Question: what does the *boundary* between positive and negative look like for Naïve Bayes?




4.5
4
3.5
3
2.5
2
4
4.5
5
5.5
6
6.5
7
7.5
8

$$\operatorname{argmax}_y \prod_i P(X_i = x_i \mid Y = y)P(Y = y)$$

$$= \operatorname{argmax}_y \sum_i \log P(X_i = x_i \mid Y = y) + \log P(Y = y)$$

$$= \operatorname{argmax}_{y \in \{+1,-1\}} \sum_i \log P(x_i \mid y) + \log P(y)$$

two classes only

$$= sign\left(\sum_i \log P(x_i \mid y_{+1}) - \sum_i \log P(x_i \mid y_{-1}) + \log P(y_{+1}) - \log P(y_{-1}) \right)$$

$$= sign\left(\sum_i \log \frac{P(x_i \mid y_{+1})}{P(x_i \mid y_{-1})} + \log \frac{P(y_{+1})}{P(y_{-1})} \right)$$

rearrange terms

$$\mathrm{argmax}_y \prod_i P(X_i = x_i \mid Y = y\ )P(Y = y\ )$$

$$= sign\left(\sum_i \log \frac{P(x_i \mid y_{+1})}{P(x_i \mid y_{-1})} + \log \frac{P(y_{+1})}{P(y_{-1})}\right)$$

if x_i = 1 or 0 ….

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

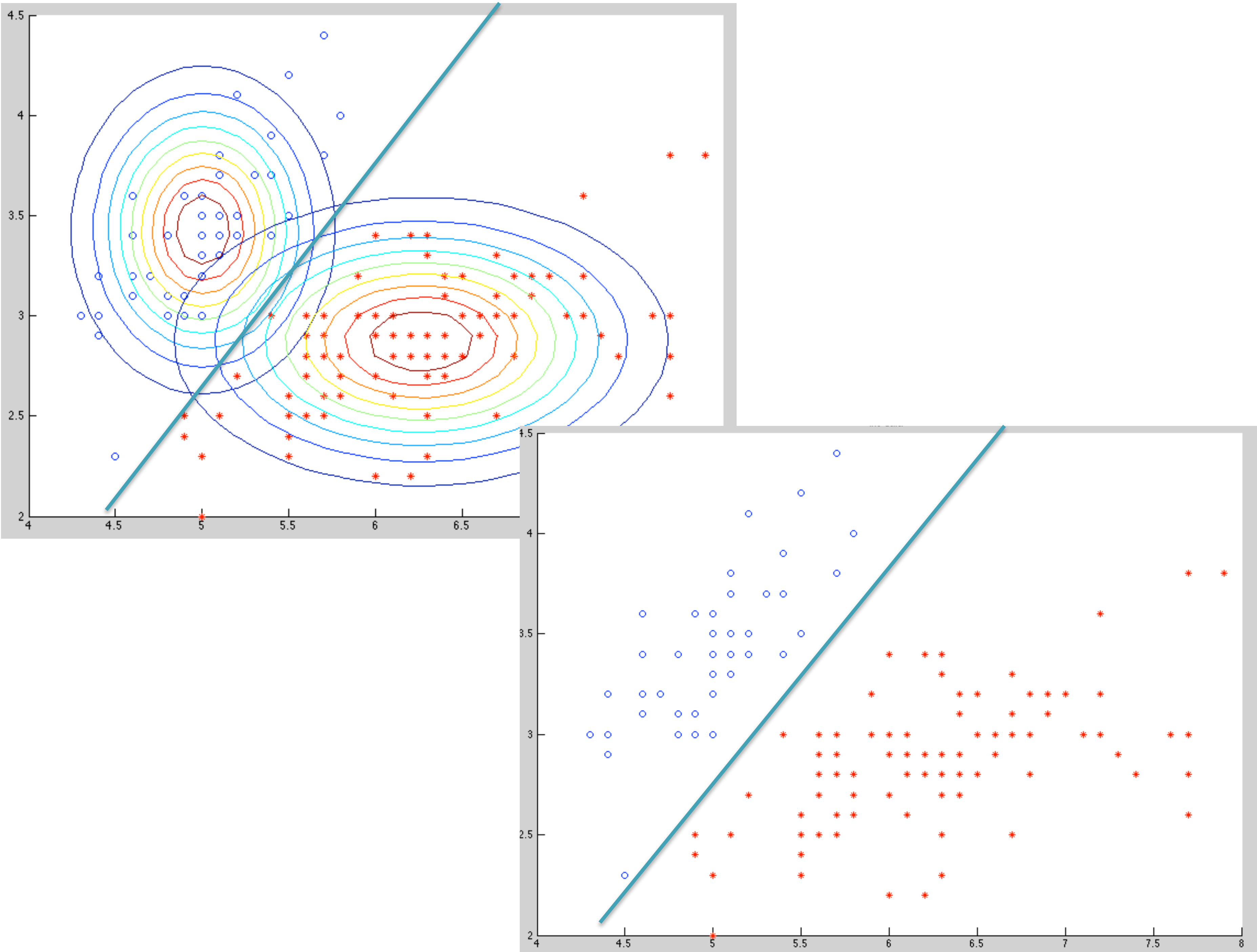
$$= sign\left(\sum_i x_i \boxed{\left(\log \frac{P(x_i = 1 \mid y_{+1})}{P(x_i = 1 \mid y_{-1})} - \log \frac{P(x_i = 0 \mid y_{+1})}{P(x_i = 0 \mid y_{-1})}\right)} + \boxed{\sum_i \left(\log \frac{P(x_i = 0 \mid y_{+1})}{P(x_i = 0 \mid y_{-1})}\right) + \log \frac{P(y_{+1})}{P(y_{-1})}}\right)$$

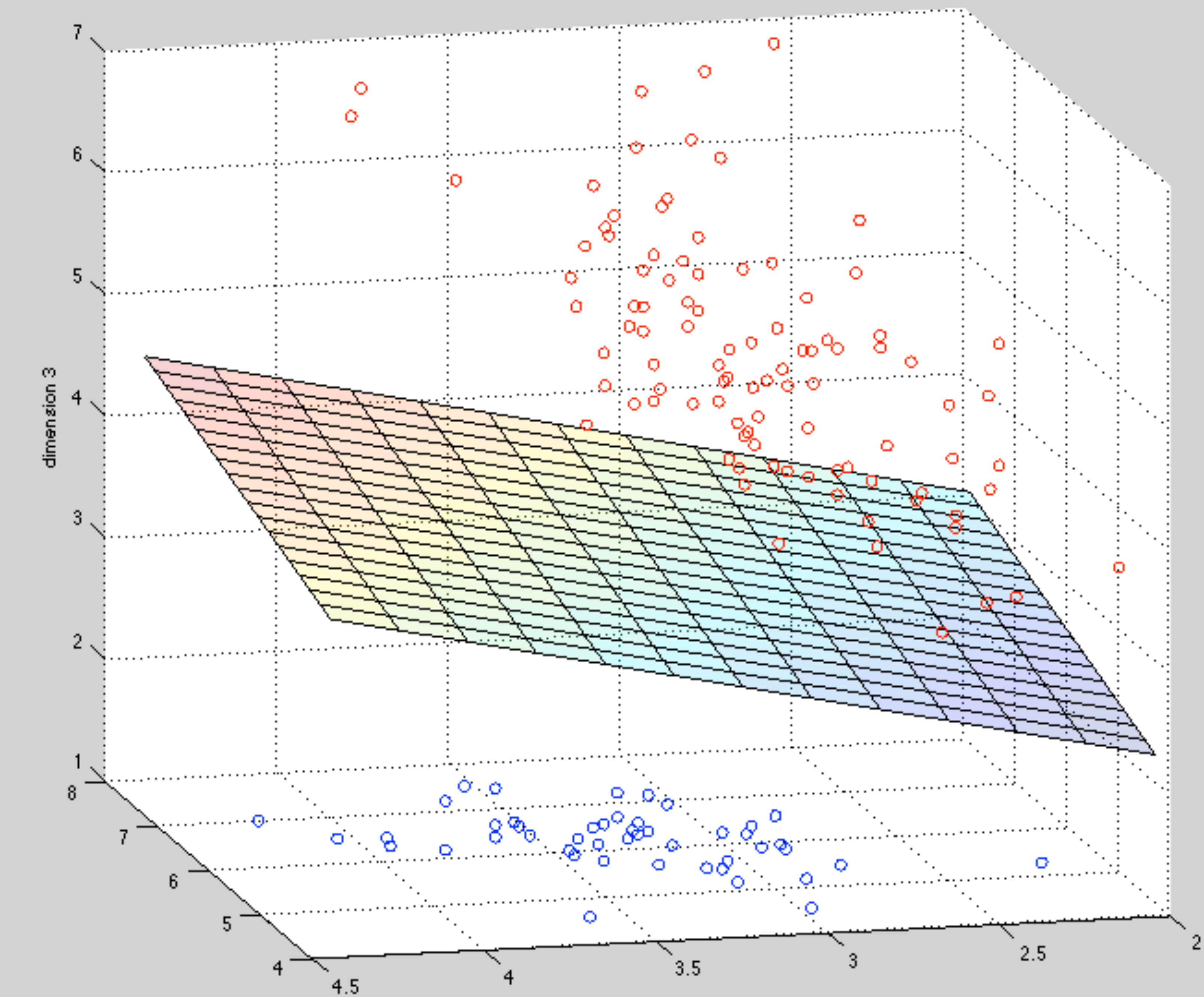
$$\boxed{u_0 = \sum_i \left(\log \frac{P(x_i = 0 \mid y_{+1})}{P(x_i = 0 \mid y_{-1})}\right) + \log \frac{P(y_{+1})}{P(y_{-1})}}$$

$$= sign\left(\sum_i x_i u_i + u_0\right)$$

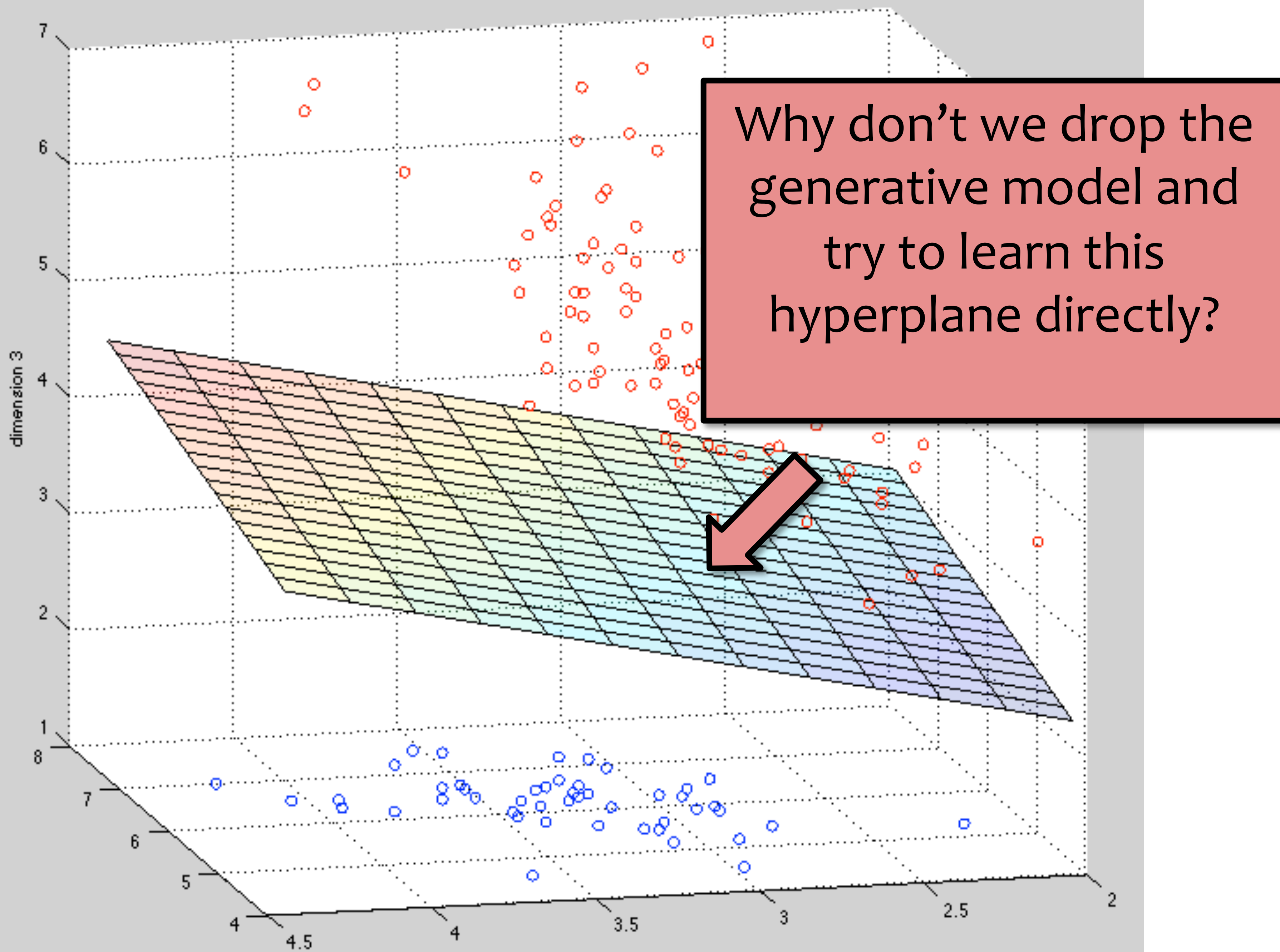
$$= sign\left(\sum_i x_i u_i + x_0 u_0\right) \quad = sign(\mathbf{x} \cdot \mathbf{u})$$

$x_0 = 1$ for every $\mathbf{x}$ (bias term)


4.5
4
3.5
3
2.5
2
4
4.5
5
5.5
6
6.5
7
7.5
8



dimension 3
7
6
5
4
3
2
1
8
7
6
5
4
4.5
4
3.5
3
2.5
2


Why don't we drop the generative model and try to learn this hyperplane directly?
dimension 3
7
6
5
4
3
2
1
8
7
6
5
4
4.5
4
3.5
3
2.5
2

Model 4: Multiclass Naïve Bayes

Model:

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

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

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

Naïve Bayes Assumption

Recall…

Conditional independence of features:

$$P(X_1, \ldots, X_K, Y) = \boxed{P(X_1, \ldots, X_K | Y)} P(Y)$$
$$= \boxed{\left(\prod_{k=1}^{K} P(X_k | Y) \right)} P(Y)$$

$$= P(X_1, \ldots, X_K|Y)P(Y)$$
$$= \left(\prod_{k=1}^{K} P(X_k|Y)\right) P(Y)$$

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!


★ ELECTION 2016 ★ MORE ELECTION COVERAGE ▸

Trump Spends Entire Classified National Security Briefing Asking About Egyptian Mummies

NEWS IN BRIEF August 18, 2016
VOL 52 ISSUE 32 · Politics · Politicians · Election 2016 · Donald Trump

- 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